Quantitative Analysis of Non-Linear Probabilistic State Estimation Filters for Deployment on Dynamic Unmanned Systems

Benjamin James McLoughlin

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

September 2019

Declaration

I, Benjamin James McLoughlin, declare that this thesis titled, **Quantitative Analysis of Non-Linear Probabilistic State Estimation Filters for Deployment on Dynamic Unmanned Systems** and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such indications, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 23/09/2019

Abstract

The work conducted in this thesis is a part of an EU Horizon 2020 research initiative project known as DigiArt. This part of the DigiArt project presents and explores the design, formulation and implementation of probabilistically orientated state estimation algorithms with focus towards unmanned system positioning and three-dimensional (3D) mapping.

State estimation algorithms are considered an influential aspect of any dynamic system with autonomous capabilities. Possessing the ability to predictively estimate future conditions enables effective decision making and anticipating any possible changes in the environment.

Initial experimental procedures utilised a wireless ultra-wide band (UWB) based communication network. This system functioned through statically situated beacon nodes used to localise a dynamically operating node. The simultaneous deployment of this UWB network, an unmanned system and a Robotic Total Station (RTS) with active and remote tracking features enabled the characterisation of the range measurement errors associated with the UWB network. These range error metrics were then integrated into an Range based Extended Kalman Filter (R-EKF) state estimation algorithm with active outlier identification to outperform the native approach used by the UWB system for two-dimensional (2D) pose estimation.

The study was then expanded to focus on state estimation in 3D, where a Six Degree-of-Freedom EKF (6DOF-EKF) was designed using Light Detection and Ranging (LiDAR) as its primary observation source. A two step method was proposed which extracted information between consecutive LiDAR scans. Firstly, motion estimation concerning Cartesian states x, y and the unmanned system's heading (ψ) was achieved through a 2D feature matching process. Secondly, the extraction and alignment of ground planes from the LiDAR scan enabled motion extraction for Cartesian position z and attitude angles roll (θ) and pitch (ϕ). Results showed that the ground plane alignment failed when two scans were at 10.5° offset. Therefore, to overcome this limitation an Error State Kalman Filter (ES-KF) was formulated and deployed as a sub-system within the 6DOF-EKF. This enabled the successful tracking of roll, pitch and the calculation of z. The 6DOF-EKF was seen to outperform the R-EKF and the native UWB approach, as it was much more stable, produced less noise in its position estimations and provided 3D pose estimation.

For their love, patience and support, this doctoral thesis is dedication to my parents:

James & Pauline McLoughlin

Contents

Acknowledgements

I would like to begin by thanking my Director of Studies Dr. Frederic Bezombes for firstly, presenting me with an opportunity to embark upon this Ph.D. journey and secondly, for his guidance, time and support during the course of my research. I would also like to thank Dr. Mohammed Kara-Mohammed and Prof. Andy Shaw for their supervision and assistance.

I express the utmost love and gratitude to all my family for their encouragement and the offering of selfless advice during my journey. An extra special and enormous thank you to Mum, Dad, Alex and James for coping with my academic related stress and mood swings on a daily basis and I hope I wasn't too hard to be around.

To my partner Maria Rebeca, thank you for your love, patience and motivation which no doubtfully spurred me on towards the finishing line. I will be forever grateful for your sacrifices which allowed us to be closer together during my Ph.D. and I look forward to what the future brings for us.

Abbreviations

2D – Two Dimensional

3D – Three Dimensional

6DOF – Six Degrees-of-Freedom

AI – Artificial Intelligence

ALS – Airborne Laser Scanning

AOA – Angle of Arrival

APD – Avalanche Photodiode

CF – Complementary Filter

CLOS – Clear Line of Sight

CR – Confidence Region

DD – Differential Drive

DOF – Degrees of Freedom

ED – Euclidean Distance

EDM – Electromagnetic Distance Measurement

EKF – Extended Kalman Filter

ERM – Error Model

ES-KF – Error State Kalman Filter

EU – European Union

FCU – Flight Control Unit

FOG – Fibre Optic Gyroscope

FOV – Field of View

GTRF – Gallileo Terrestrial Reference Frame

GNSS – Global Navigation Satellite System

HDL – High Definition LiDAR

ICR – Instantaneous Centre of Rotation

IMU – Inertial Measurement Unit

IKF – Indirect Kalman Filter

INS – Inertial Navigation System

KF – Kalman Filter

LED – Light Emitting Diode

LiDAR – Light Detection and Ranging

LLS – Linear Least Squares

LQE – Linear Quadratic Estimator

MAV – Micro Aerial Vehicle

MBE – Mean Bias Error

MCL – Monte Carlo Localisation

MCU – Micro-Controlling Unit

MI – Mutual Information

MVO – Monocular Visual Odometry

MD – Mahalanobis Distance

MM – Markov Model

MLS – Mobile Laser Scanning

NCLOS – Non-Clear Line of Sight

NED – North East Down

NLM – Non-Linear Model

NDT – Normal Distribution Transformation

NTP – Network Time Protocol

PDF – Probability Density Function

PLR – Portable Laser Ranger

PMF – Probability Mass Function

POV – Point of View

R/C – Remote Control

R-EKF – Range Extended Kalman Filter

RANSAC – Random Sampling Consensus

RF – Radio Frequency

RFID – Radio Frequency Identification

RMSE – Root Mean Squared Error

ROS – Robot Operating System

R-PI – Raspberry Pi

RTC – Real Time Clock

RTK – Real-Time Kinematic

RTS – Robotic Total Station

SLAM – Simultaneous Localisation and Mapping

SS – Skid Steering

SURF – Speeded Up Robust Features

SVO – Stereo Visual Odometry

TDC – Time-Digital Converters

TIA – Transimpedance Amplifier

TOA – Time of Arrival

TOF – Time of Flight

TLS – Terrestrial Laser Scanning

UAS – Unmanned Aerial System

UAV – Unmanned Aerial Vehicle

UWB – Ultra-wide band

VLOAM – Visual-LiDAR Odometry and Mapping

VLP – Velodyne Lidar Puck

VO – Visual Odometry

VR – Virtual Reality

WAP – Wireless Access Point

WGS-84 – World Geodectic System 84

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction

1.1 Research Motivations

The technological advancements made in modern day data capture systems, techniques and methodologies has enabled rapid processing of information to allow consumers to better understand scenarios, to enhance traditional operation procedures and/or to make crucial decisions. The work carried out within this thesis is based on a modern day data capture system in the form of an unmanned robotic system and how the design of such a complex system was utilised within an European Union (EU) Horizon 2020 project known as DigiArt (Grant No. 665066). The DigiArt project consisted of seven consortium partners spanning across five European countries whose overall objective was to create virtual reality (VR) environments of two historical locations. Such influential locations included the subterranean Neanderthal Sclayn Cave in Belgium and the Palace of Aigai in Northern Greece. The opportunity to provide virtual tours to those who were unable to regularly visit such influential archaeological sites is what made the DigiArt project both innovative and effective. In order to create these virtual tours, there was a necessary link between the actual physical sites and their virtual representations, therefore, unmanned systems with remote sensing capabilities were deployed. The remote sensing aspects of such unmanned systems was through the utilisation of an optical three-dimensional (3D) measurement technique known as Light Detection And Ranging (LiDAR). The recent technological advancements achieved in the simultaneous development and deployment of mobile unmanned systems and compact miniature payloads such as LiDAR devices has revolutionised the approaches that research and industrial institutions are utilising. The ability to design sophisticated control system architectures to provide an unmanned agent with a degree of intelligence and

autonomy through the implementation of learning, classification and artificial intelligence (AI) platforms has resulted in an increase in productivity and objective completion efficiency. However, prior to the integration of intelligent platforms, an influential and foundational aspect of an autonomous unmanned agent is its ability to answer a key question: What is my current state? The fundamentals of this question heavily relate to a reputable component of mobile robotics known as state estimation, which is paramount when assessing the unmanned system's current state within an operational environment. In order to achieve autonomy, the unmanned system is required to answer such a question with as little uncertainty as possible. This situation is no different when operating in the archaeological environments as a part of the DigiArt project.

The work conducted within this thesis assesses the problem of unmanned and mobile robot localisation, with the application towards 3D mapping and environment reconstruction as a main deliverable for the DigiArt project Work Package 3. The experimental and theoretical content produced was a combination of statistical robotic implementations as well as sensory system analysis, where sensors were influential as they enable an unmanned system's perception of its environment. Intelligent mobile unmanned systems are independent to the observation of human interaction, therefore they solely rely on the perception of their current operating environment through the utilisation of the on-board sensing systems. For example, autonomous robotic systems in automotive product design and assembly lines are executing tasks at astounding rates and improving productivity and reducing time. The general nature of these systems, however, is to function statically in confined, predictable and controllable operating environments reducing the capabilities of the system to adjust to unpredictable situation occurrences. Mobile robotics, contrastingly, operate in unstructured and uncontrollable environments, therefore trusting the reliability of the on-board sensing systems becomes paramount to providing optimal estimations of the unmanned system's temporal state transitions. The integration of external sensor observations with coherent system identification through mathematical modelling is what forms the basis of estimation theory, primarily the probabilistic approach through the application of Bayesian statistical interpretations. Therefore, this thesis presents the integration of statistical and probabilistic robotic techniques with an aim to develop adaptive and effective state estimation systems that enabled an unmanned system to generate detailed maps of its surrounding environment. The overview and structure of

this thesis is presented in the following section.

1.2 Thesis Structure & Overview

This thesis consists of seven chapters and is divided into four main parts presenting the stages undertaken in the design and development of mathematical models and control algorithms for recursive state estimation of an unmanned system. The structure and overview of each part and chapter is as follows:

- Part I - Introduction

  Part I of this thesis comprises of two overall chapters. These chapters are constructed to provide an introduction to the thesis as well as the theory based on the topics that where highly influential in the design of frameworks, methodologies and systems utilised throughout this thesis.

  - Chapter 1 - Introduction

    Provides an introduction to the thesis and covers the research motivations, primary aims and objectives and the contributions to knowledge that the work in this thesis provides.

  - Chapter 2 - Background Theory

    Chapter 2 follows with the presentation of the relevant theoretical content applied throughout this thesis. This includes the introduction to the concepts of probability, Bayesian inference and the fundamental aspects of sensor fusion and recursive state estimation for mobile robotic systems.

- Part II - Unmanned System Design & Mathematical Modelling

  Part II is segmented into three chapters. It begins with the overview of the unmanned system design and its mathematical modelling based on the kinematic nature. Part II also presents the methods implemented to assess robot state estimation in both two and three dimensions (2D, 3D).

  - Chapter 3 - Robotic System Design

    Chapter 3 provides details concerning the custom mobile robotic systems that where designed and deployed for algorithm implementations for this work.

- Chapter 4 - 2D Localisation Problem

  Chapter 4 demonstrates a framework for how the correct approach to quantifying sensor measurement error is influential in the design of recursive state estimation algorithm for mobile robotics. This begins with the simplified assessment of the problem in a 2D model, where a 2D pose of the unmanned system is studied. Additionally, Chapter 4 investigates the sensor uncertainty characterisation methodology that is considered as a novel implementation.

- Chapter 5 - 6-DOF Motion Estimation

  Post completion of Chapter 4, the problem assessment is then advanced into state estimation in 3D space and presented within Chapter 5. This includes modelling the system in six main degrees-of-freedom (6-DOF). The kinematic nature of unmanned systems enable the platform to manoeuvre in 3D, this is also the case for ground vehicles due to topology changes. Therefore, it was necessary to also assess the problem in 3D. An algorithm is provided which uses data acquired from a navigational LiDAR system fused with an inertial measurement unit (IMU) to estimate unmanned system 6-DOF pose.

- Part III - Algorithm Validation

  Part III presents the quantitative analysis undertaken to compare each designed algorithm and possible positioning solution against the ground truth metrics.

  - Chapter 6 - Quantitative Analysis

    Chapter 6 discusses the quantitative analysis procedure on the possible positioning solutions using ground truth metrics as a method to quantify errors.

- Part IV - Conclusions

  Part IV presents conclusions regarding the key conclusions of the work presented in this thesis as well as ideas for future developments.

  - Chapter 7 - Conclusions & Future Work

    Chapter 7 presents the conclusions of the work conducted in this thesis and possible ideas to be implemented in order to further develop the system design.

1.3 Aims & Objectives

The main aim of this work was to demonstrate how the effective characterisation of sensor measurement variance can optimise state estimation procedures for unmanned systems. The individual objectives to achieve the design of a framework to achieve the main aim were;

- Design of a remotely operated unmanned experimental platform capable and robust enough to carry the required payloads whilst operating in both rugged and non-rugged environments.
- Generate a mathematical motion model of the unmanned experimental platform based on its kinematic nature and its on-board IMU.
- Design the dynamic sensor error variance characterisation framework for the positioning sensors that were set to operate on-board the unmanned platform (referred to as the ground truth metric).
- Investigate the error variance of a 2D ultra-wideband (UWB) range- based measurement system using the dynamic ground truth metric as reference.
- Design a Range-based Extended Kalman Filter (R-EKF) state estimation algorithm incorporating the motion model and raw range error variance of a UWB system.
- Progress onto 3D state space with algorithm design utilising a 3D LiDAR system and an orientation Error State Kalman Filter (ES-KF) to provide measurements concerning 6DOF pose of the unmanned system.
- Quantify and examine error variance outputs from 6DOF pose estimations from the LiDAR algorithm and ES-KF using a set of static and simulated ground truth metrics.
- Design of a 6DOF-EKF integrating motion model and error variance from the LiDAR and ES-KF.
- Examine the performance of the R-EKF and 6DOF-EKF in comparison to the traditional techniques and dynamic ground truth metrics.

1.4 Contributions

This thesis presents a systematic methodology for the design and implementation of robotic system state estimation control with the application towards localisation and mapping. The approach taken identifies the uncertainty within readings acquired from positioning sensors for unmanned systems using a surveying grade optical tracking system as the ground truth metric. Such uncertainties are then used to implement sensor fusion algorithms to optimise the estimated states of the unmanned system as it navigates within an environment. In order to demonstrate the effectiveness of the approach, each underlying element within the overall designed system is successfully tested and compared to the ground truth metric. Therefore, it is shown that the application of mathematical kinematic modelling, robotic optimal control and estimation provide an effective systematic framework for the development of an unmanned mobile 3D mapping system for subterranean environments.

1.4.1 Uncertainty Characterisation of Unmanned System Positioning Techniques & Implementation within Bayesian Probabilistic State Estimators

This area of novel contribution focuses on the structured methodology undertaken to characterise the uncertainty of the positioning systems operating on-board an unmanned mobile ground rover. This methodology then includes the use of this uncertainty metric to formulate and implement an EKF state estimation algorithm. The proposed methodology utilised a Robotic Total Station (RTS), which is a surveying grade optical instrument to actively track the unmanned system providing a translational ground truth metric. The output readings from the mobile agent's positioning system are then compared to the ground truth to acquire an uncertainty reading. This uncertainty is then used to design a range-based variant of an EKF, where the resulting position estimations are seen to be an improvement in comparison to the estimations produced by the native positioning system's algorithm. The following summary highlights the key contributions which the presented work concerning this topic has made to the field:

- Demonstrates how an RTS can be used to characterise the uncertainty of a UWB sensor network that is used as a positioning solution for unmanned systems.
- Determines how the uncertainty metrics acquired using the RTS can be integrated

into the formulation process for an EKF measurement model.

- Conducts a validation process to prove that accurately measuring the uncertainty of mobile robot positioning systems can improve the precision of position estimations through the implementation of a R-EKF in comparison to the standard odometry motion model and the native UWB localisation algorithm.

1.5 Publications

This section lists the peer reviewed and published studies conducted as a part of this thesis.

1.5.1 International Journal Articles

- B.J.McLoughlin, H.A.G.Pointon, J.P.McLoughlin, A.Shaw & F.A.Bezombes. *Uncertainty Characterisation of Mobile Robot Localisation Techniques using Optical Surveying Grade Instruments.* Sensors 2018, 18(7), 2274; https://doi.org/10.3390/s18072274

- H.A.G.Pointon, B.J.McLoughlin, C.Matthews & F.A.Bezombes. *Towards a Model Based Sensor Measurement Noise Input for Extended Kalman Filter State Estimation.* Drones 2019, 3(1), 19; https://doi.org/10.3390/drones3010019

1.5.2 International Conferences

- B.J.McLoughlin, J.Cullen, A.Shaw & F.A.Bezombes. *Towards an Unmanned 3D Mapping System using UWB Positioning.* Towards Autonomous Robotic Systems (TAROS) 2018. Lecture Notes in Computer Science, Vol 10965. Springer, Cham

1.5.3 Horizon 2020 Project Dissemination Reports

- *Initial Report on Static & Aerial Scanner*, DigiArt Project Work-Package 3.2 -, http://digiart-project.eu/project/reports/d3-2-report-on-the-static-and-aerial-sensors/

- *First Version of Static & Aerial Scanner*, DigiArt Project Work-Package 3.3, http://digiart-project.eu/project/reports/d3-3-first-version-of-the-aerial-and-static-scanning-sensor/

1.5.4 Internal Faculty Publications

- B.J.McLoughlin. *Sensor Roll & Pitch stabilisation for an UAV-based LiDAR system.* Liverpool John Moores University Faculty of Engineering & Technology Research Week 2018. Proceedings accessed at: https://www.ljmu.ac.uk/∼ /media/files/ljmu/research/centres-and-institutes/engineering-and-tech/fet_frw_proceedings_2017.pdf?la=en

Chapter 2

Foundational Theoretical Concepts

Chapter 2 presents the literature concerning the fundamental topics of study utilised throughout this thesis. As presented in Chapter 1, the main aspects of the experimental procedures conducted at each stage of this thesis are heavily related to the topic of sensor fusion and robotic state estimation through the utilization of the EKF algorithm. It is within this chapter, that the foundational aspects of recursive robotic state estimation and the EKF algorithm itself are derived and explained through statistical probabilistic theory, and how the aforementioned aspects are integrated within state space modelling procedures to enable the manipulation of 2D and 3D geometric systems. The main aim of Chapter 2 is to provide an insight into the theoretical topics required prior to progressing through this thesis. Within each topic of literature, graphical overviews are used to best explain probability distributions and general factors relating to stochastic systems.

2.1 Concepts of Probability

The theory presented in this section concerns the governing fundamental aspects of probability theory that directly relate to, or influence, the topic of recursive state estimation for mobile robotic systems. It is important to note here that the topic of recursive state estimation for robotics is built on the basic concepts of random variable analysis, conditional probability theory and the progression into Bayesian statistical modelling and influence. In the following literature, the aforementioned areas of mathematics are presented to provide outlines concerning their operations, rules and formulations.

2.1.1 Random Variable Analysis

The execution of a set of random experiments or a sequence of system state transitions provide outcomes that can take up many forms, therefore it is a methodical approach to associate numbers to each outcome, such associates are defined as *random variables* [1] [2]. For presentation simplicity, the denotation $\boldsymbol{X}$ is set to represent a randomly operating variable and x as a state that $\boldsymbol{X}$ may adopt at any certain instance of time. A perfect example of a random variable is the integer rolled on a dice, where this variable meets the condition $x \in [1 \dots 6]$. Characterising the outcome of this random variable $\boldsymbol{X}$ is solely dependent upon whether the nature of $\boldsymbol{X}$ is discretely or continuously operating. A discrete random variable is defined as a variable which has a finite or countable range, the scenario of rolling a dice being a prime example. However, the contrasting continuous random variable is defined as one that adopts values within an uncountable or infinite range [3] [4]. The conditions shown in Equation (2.1) and Equation (2.2) best portray the difference between a discrete and a continuous random variable.

$$p(x) = P(\boldsymbol{X} = x) \tag{2.1}$$

$$p(x) = P(\boldsymbol{X} = x) = 0 \tag{2.2}$$

The condition shown in Equation (2.1) for a discrete random variable, states that the Probability Mass Function (PMF) $p(x)$ gives the probability that the random variable $\boldsymbol{X}$ will adopt a state x. This is not the case for a continuous random variable as shown in Equation (2.2), as it states that the probability that $\boldsymbol{X}$ will adopt an exact state of x is equal to zero [5]. The vital aspect to understand from Equation (2.2) that represents the probability conditions for a continuous random variable is due to the nature of its infinite range, it is therefore only possible to calculate the probability that $\boldsymbol{X}$ belongs to a certain interval. In the case of mobile robotics, variables such as the pose of the system will be estimated continuously and more often than not, the state of the environment has a great influence on the transition of such variables. This is an important factor to note as environmental conditions are most likely to be unpredictable, therefore the state that variables such as the pose of a robotic system change will adapt, will be within an infinite

or uncontrollable set space [6]. Therefore, from here onwards continuous random variables will be considered.

The probabilities of a continuous random variable, in this case $\boldsymbol{X}$, are calculated using a Probability Density Function (PDF). It is to note, however, that the PDF itself does not provide probabilities [1]. To present this more effectively, an example univariate normal distribution of a continuous random variable is generated using the standard univariate normal distribution formulation shown in Equation (2.3), where μ and σ denote the mean and standard deviation moments of the distribution respectively and were set to $\mu = 0$ and $\sigma = 1$.

$$p(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{2.3}$$

The method undertaken to calculate the probabilities of continuous random variables is analysing the area under the curve generated from the PDF within certain intervals [7]. The blue highlighted area shown in Figure 2.1 shows the probability density of $\boldsymbol{X}$ adopting x within the infinite range interval, this satisfies the condition shown in Equation (2.4).

$$\int_{-\infty}^{\infty} p(x)dx = 1 \tag{2.4}$$

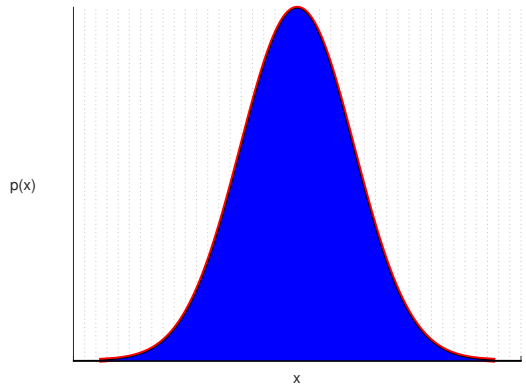


Figure 2.1: Probability density within a finite interval

The calculation of the probability density is when the area under the curve $y = p(x)$ is assessed within the infinite range [8]. However, isolating a certain sub-interval enables the calculation of the probability of $\boldsymbol{X}$ adopting a value within the designated sub-interval. The calculated probability is shown by the shaded blue area in Figure 2.2 and satisfies the conditions shown in Equation (2.5)

$$\int_a^b p(x)\ dx = P(a \leq \boldsymbol{x} \geq b) \tag{2.5}$$

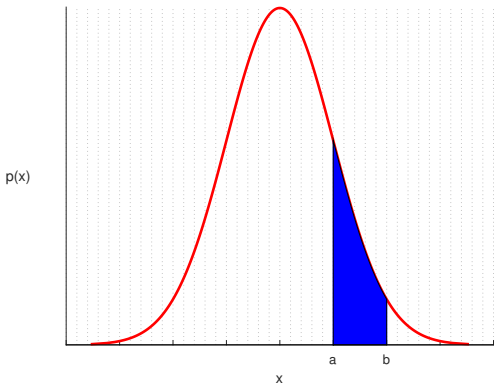


Figure 2.2: Probability of continuous variable being within a sub-interval

Therefore, for confirmation purposes, using Equation (2.5) to assess the condition $y = p(x)$ for the probability of the continuous variable $\mathbf{X}$ to adopt an exact value, the result of this is shown in Equation (2.6) and thereby meeting the condition previously presented in Equation (2.2).

$$P(\mathbf{X} = a) = P(a \leq \mathbf{X} \geq a) = \int_a^a p(\mathbf{X} = x)\ dx = 0 \tag{2.6}$$

In the topic of state estimation for mobile robotic systems, PDFs are used to represent the likelihood of a variable relating to the system $\boldsymbol{X}$ being at state x within a certain interval, given the acquisition of an external observation measurement source. The acquisition of external data sources gathered from sensory systems on-board the mobile robot result in the integration of a conditioning variable. This is covered in the following section that focuses on joint and conditional probability.

2.1.2 Joint & Conditional Probability Densities

Section 2.1.2 provides an overview into the theory concerning joint and conditional probabilities and their corresponding density functions.

2.1.2.1 Joint Probability Densities

A joint probability density is one which considers an outcome relating to more than one random variable [3], for example in a statistical robotics scenario, the relative pose of a ground-based unmanned system, which consists of three variables that may react randomly

to changes in the environment. These variables are the Cartesian position (x, y) and the system's relative heading, (ψ). These variables will have their own individual distributions, however, there may be some circumstances where the probability combinations of these variables may be required. In these cases, the probability combinations are represented through a joint PDF $p(x, y)$.

2.1.2.2 Conditional Probability Densities

A conditional probability consists of the calculation of a probability of a random variable given that some prior knowledge relating to the outcome is known [1]. For example, relating back to the robotics scenario, the probability calculation of the next update of the robot's heading given that there has been an observation update from the on-board sensors. The conditional probability is represented as $p(x|y)$.

2.1.2.3 Joint & Conditional Probability Relationship

A joint probability density of the occurrence of both $\boldsymbol{X}$ and $\boldsymbol{Y}$ adopting states x and y represented as $p(x, y)$ can be factored into conditional and non-conditional components. The conditional probability component assesses the occurrence of y given event x, $p(y|x)$ and the reverse conditional probability $p(x|y)$. This is shown in Equation (2.7) and Equation (2.8), where the individual and non-conditional probability components of each event are represented as $p(x)$ and $p(y)$.

$$p(x, y) = p(y \mid x)\, p(y) \tag{2.7}$$

$$p(x, y) = p(x \mid y)\, p(x) \tag{2.8}$$

The formulations of the joint probability densities shown in Equation (2.7) and Equation (2.8) can then be solved for $p(x, y)$ giving that shown in Equation (2.9).

$$p(x \mid y)\, p(y) = p(y \mid x)\, p(x) \tag{2.9}$$

The condition shown in Equation (2.9) is then used to formulate what is known as the Bayes' rule of probability. In robotics, probability densities are assessed within a Bayesian framework [8]. It is worth noting, following on from the initial definitions of conditional

probability and the declaration of Equation (2.9), it is important to present the theorem of total probability for a continuous case which is shown in Equation (2.10). Equation (2.10) is an important formula implemented in the design of algorithms applied to mobile robotic systems, which are presented further on.

$$p(x) = \int p(x \mid y)\, p(y)\, dy \tag{2.10}$$

The following section therefore explores the topic of Bayesian probability and its influence in estimation systems for mobile robotics.

2.2 Bayesian Probability

The theory of Bayesian probability presents a mathematical framework for performing the process of statistical interference, where such process is executed following the laws of the *Bayes* probability rule which is generalised as a manipulation of current conditional probabilities [9][10].

Equation (2.9) can then be rearranged to arrive at the Bayes' rule of probability shown in Equation (2.11) and Equation (2.12).

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} \tag{2.11}$$

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} = \frac{p(y \mid x)p(x)}{\int p(y \mid x)p(x)} \tag{2.12}$$

To simplify the presentation of Bayesian theory, nomenclature is presented from now onwards using $\boldsymbol{H}$ to define a hypothesis at hand and $\boldsymbol{O}$ as the set of data retrieved concerning observations of the hypothesis $\boldsymbol{H}$. Additionally, notations t are used to denote time instances to demonstrate how the beliefs of the hypothesis are updated. The process involved in statistical Bayesian inference is updating one's initial beliefs about a hypothesis given the observation of retrievable data, $P(H_{t+1}|O_t)$ [11]. This framework allows one to formally incorporate prior knowledge or beliefs of the hypothesis, $P(H_t)$, with the probability density or likelihood of the observations given the initial belief of the hypothesis, $P(O_t|H_t)$. Therefore, re-presenting Equation (2.11) in this format is demonstrated in **??**, where $P(O_t)$ is the marginal likelihood in this case represented as the total likelihood of

the observations over H_t and H_{t+1} as shown in Equation (2.13).

$$p(H_{t+1}|O_t) = \frac{p(O_t|H_t)p(H_t)}{p(O_t)} \tag{2.13}$$

Section 2.2 presented an overview of Bayesian inference and probability theory. The inclusion of this topic is influential in the formulation of the predictive algorithms used in the design of recursive robotic state estimation systems. Therefore, the following section introduces the foundations of recursive state estimation for mobile robotics. It firstly introduces fundamental aspects of recursive state estimation and its relation to state-space modelling. It then discusses how Bayesian probability and general probability concepts that were previously presented are integrated with state estimation for mobile robotics.

2.3 Recursive Robotic State Estimation

The topic of state estimation is considered the core area of the probabilistic analysis approach to mobile robotic systems [12]. State estimation addresses the issue of estimating variables specific to a system that are not directly measurable through the inclusion of an observation source from on-board sensory units. It is also possible, however, to provide improved estimations concerning already measurable variables through the fusion of multiple sources of sensor observations [13]. The operating environments that unmanned systems are subjected to cannot directly be controlled and if a system is to be robust and easily deployable, it has to possess the capability of adapting to such dynamic environments, therefore the only method for an unmanned system to achieve this environmentally adaptable characteristic is through the observations provided by the on-board sensors [14]. However, sensor observations in general are subjected to perturbation effects leading to limitations in what sensors can perceive, this results in the generation of an error quantity within the measurement. Therefore, probabilistic state estimation approaches towards mobile robotic systems utilise Bayesian based frameworks to incorporate this error metric and probabilistic mathematical models to provide optimal estimations of the states concerning the unmanned system. Section 2.3 therefore presents the fundamental topics of recursive state estimation prior to the presentation of how it relates to the rules of Bayesian probability. Such topics are outlined in the following sections.

2.3.1 Uncertainty

The topic of uncertainty is a vital component of robotic systems, technology and research applications. As mentioned in the introduction to Section 2.3, unmanned systems rely on their on-board sensors to interact and perceive the environment, enabling the system to then judge and make decisions on how to proceed. However, observations from sensors experience erroneous readings due to perturbation effects. This may be caused by environmental factors or faults relating to the manufacture of the sensor devices. Erroneous sensor observations are categorised as an aspect of *uncertainty* that contributes towards the unmanned system as a whole. Uncertainty is used as a global term and it arises when the system experiences a lack of information that is critical to its performance [15]. There are multiple possible factors that influence uncertainty in mobile robotics, where the most important of such factors are outlined below [12].

- Environments: As mentioned previously, environmental conditions are generally uncontrollable and unpredictable unless a system is constrained in its design to be dependent on and operate within a particular environment. Occurrences within environments such as dynamic objects or features, volatile terrains are all aspects which effect the uncertainty levels of the design system as it will have to adapt and decide on its next decision.

- Sensors: Observations acquired from sensors are accompanied with a level of error or noise. As mentioned above, sensors are limited by what they can perceive. For example, consider an unmanned system using a laser mapping navigational system that uses features in the environment for localisation. For environments that are sparse and low in features, the performance of this navigational system example in this specific environment will be limited.

- System Models: The process of formulating models of the unmanned systems in question concerning their dynamic behaviour is another important factor considered in recursive state estimation. The generation of these models acts as a maximum likelihood outcome of a state as it temporally transitions. However, if these models fail to represent the system correctly, an increase in uncertainty will arise as errors propagate due to model inaccuracies. An example of this is modelling the kinematics

of a unidirectional unmanned ground vehicle as an omnidirectional system.

2.3.2 System Identification

Concerning the area of mobile robotics, unmanned platforms that are designed to operate as a tool for data collection and exploration can be defined as dynamic systems. With the heavy link between probability theory and recursive state estimation for robotics previously discussed, the variables related to such dynamic systems result in the system being assessed as a stochastic process, more specifically, a random process with a collection of random variables (system states) [16]. Therefore, modelling the system's behaviour prior to implementation along with its expected input and output factors is a key beneficial factor in the design of state estimators. System identification deals with the problem of generating mathematical models of dynamic systems based on observations provided by the system's on-board sensory units [16]. For example, a mathematical model of motion for an unmanned aerial vehicle (UAV) may focus on the holonomic kinematic aspects of the UAV based on the observation feed from the Global Navigation Satellite System (GNSS) and flight control unit (FCU). The importance of system identification processes is influential in the design of algorithms tasked with estimating the states related to an unmanned system. This importance stems from having full control of the system and having the flexibility to identify any irregularities, abnormalities or errors that may arise within the system during operational procedures. For example, consider a mathematical motion model of an unmanned system that was derived using proprioceptive input to estimate certain system states. If this model shows a large difference in estimations in comparison to a sensory observation source, this could aid in the identification of sensor failure as the sensor is not providing outputs similar to the motion model that is bound by physical laws [16]. It is noted that the system identification process is defined as generating mathematical models of a system based on measured observations. In this case, the mathematical models of unmanned systems used in this thesis are generated from kinematic laws and are not achieved through any regression techniques. Therefore, factors related to the work in this thesis that concern the topic of system identification are presented below.

2.3.2.1 State

When modelling an environment, a dynamic system or any system in general, a state is a variable that is characterised as having an impact on the system's future condition. States are generally classified into two categories; dynamic or static. Dynamic states experience a temporal transition, whereas static or non-dynamic states do not change. When concerning a mobile unmanned system, a state is a representation of a variable relating to the unmanned system itself. These variables include examples such as pose, both linear and angular velocity, sensor biases and other variables that may affect the future state of the system. In order to describe the transition of dynamic states over time, the generation of a mathematical model that best represents the nature of the state's transitional behaviour is utilised. These models can relate to the unmanned system itself through proprioceptive sensors such as wheel odometers and IMUs or exteroceptive sensors such as cameras, sonar and laser-based systems. Such states are used in models which best represent the transition of these states ahead of time. This is discussed in the following sections [17].

2.3.2.2 Motion Model

In robotics, a motion model is a mathematical representation of a dynamic system's kinematic behaviour. The motion model, sometimes known as a state transition function, utilises a first order differential as a transfer function that best describes the temporal transition of system states to the next iteration within the system's operating cycle [18]. It is within this process that system constraints bound by the laws of physics can be included to simplify the modelling of the system. As discussed in Section 2.3.1 concerning uncertainties within mobile robotics, a degree of uncertainty arises within the motion model that has to be included within the modelling process. An overview of the motion model and the system cycle can be seen in Figure 2.3.

2.3.2.3 Observation Model

Following on from the description of a motion model, the observation model is used to map the outputted state vector from the motion model into the same format as the observations acquired from external sensory units. In predictor-corrector systems, the beneficial aspect of this is the ability to calculate a residual between the two models. This enables close

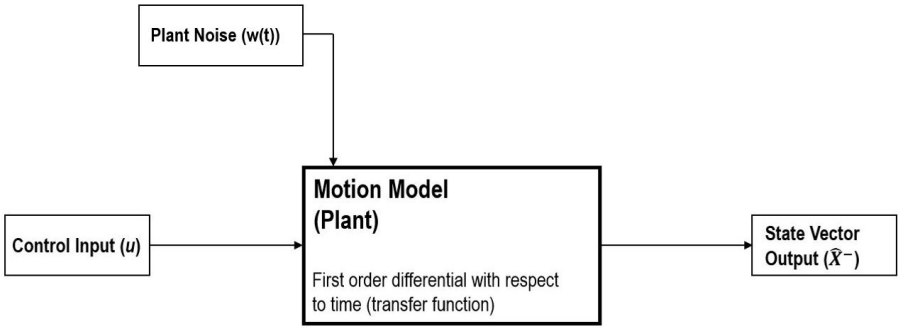


Figure 2.3: Linear Motion Model

monitoring of the system's performance and whether sensors are operating as expected. An overview of the observation model process is shown in Figure 2.4.

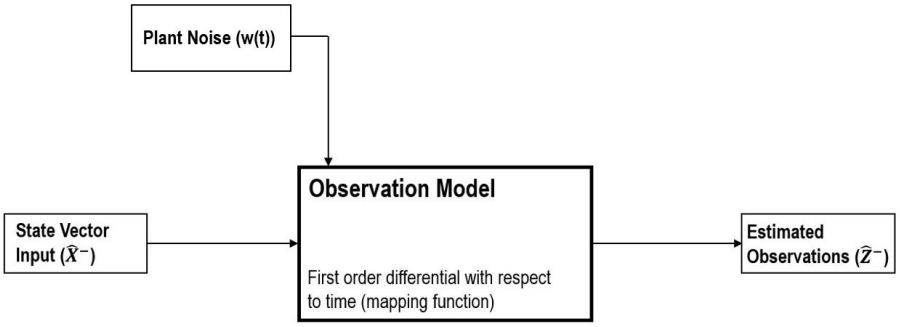


Figure 2.4: Linear Observation Model

2.4 Bayes Filter

This section presents the Bayes filter, which applies the Bayes rule of probability previously presented in Equation (2.11) into the derivation of the algorithm for recursive state estimation. It also demonstrates how the motion and observation models presented in Section 2.3.2.2 and Section 2.3.2.3 are utilised in a predict-correct process such as the Bayes filter. This section includes an overview of the Bayes filter followed by the mathematical derivation.

2.4.1 Overview

Bayes filters probabilistically estimate a dynamic system's temporally transitioning states from external and noisy sensor observations [12]. Bayes filters represent the states of the system at time t as a set of random variables x_t. For each instance in time, a

probability distribution of x_t is denoted as the belief, $Bel(x_t)$, representing the uncertainty of that current estimate of x_t [5]. A belief is an object which represents the unmanned system's internal knowledge of its states, whether they are specific to itself or its operating environment. For the description of the Bayes filter and probabilistic robotics in general, beliefs are represented through conditional probability distributions [8]. The Bayes filter is considered as a recursive process, where $Bel(x_t)$ is calculated from $Bel(x_{t-1})$. There are two main stages of the Bayes filter which make up the structure of the algorithm. These stages are defined as the control update or prediction stage and the observation update, otherwise known as the correction [8, 12]. These are discussed below.

- **Prediction**: The prediction stage of the Bayes filter uses the control input, u_t, into the system as a source of information to predict the transition of x_t. This is achieved by calculating the belief, $\overline{Bel}(x_t)$, that the system will obtain a state x_t based on the prior state belief x_{t-1} and the control input u_t. The notation of the overline within $\overline{Bel}(x_t)$ shows that this belief represents the state based on the previous state posterior. This is acquired prior to the incorporation of the external observation information sources. This is demonstrated in Equation (2.14).

$$\overline{Bel}(x_t) = \int p(x_t \mid x_{t-1}, u_t)\ Bel(x_{t-1})\ dx_t \tag{2.14}$$

  Referring to Equation (2.14), the belief $\overline{Bel}(x_t)$ is achieved by the integral of the product of two individual distributions [9]. These distributions are the probability that the occurrence of the control system input u_t results in the transition of x from time instances $t-1$ to t and the prior belief distribution of x_{t-1}. In relation to mobile robotics, the prediction stage utilises the process shown in Figure 2.3 [19].

- **Correction**: The correction stage of the Bayes filter calculates the updated belief $Bel(x_t)$ using the probability of the external observations and the posterior belief $\overline{Bel}(x_t)$. This is shown in Equation (2.15) [20].

$$Bel(x_t) = \eta\ p(z_t \mid x_t)\ \overline{Bel}(x_t) \tag{2.15}$$

  It is to be noted that the product of external measurement probability $p(z_t \mid x_t)$ and $\overline{Bel}(x_t)$ outputs a result that is not a probability. Therefore, this section of the

Bayes filter is normalised through the inclusion of a normalisation constant η. Once again, in relation to mobile robotics, the correction stage utilises the process shown in Figure 2.4

To provide a representation of the Bayes filter and belief propagation, an example is presented. This scenario provides a three-step graphical representation of the prediction and correction process as shown in Figure 2.5 [20].

The first step shown in Figure 2.5a shows that as the robot moves using solely its motion model and transition estimation from u_t, the state uncertainty is seen to grow. This is shown by the increasing width of the probability distribution from x_1 to x_3.

During the second step shown in Figure 2.5b, the robot queries its sensing systems to acquire an observation of a pillar at that current time instance. This gives a second estimation of the current position represented by the magenta distribution.

The third and final step shown in Figure 2.5c sees the robot correct its position estimation through the fusion of the belief prior to the observation (Step 1) and the probability of making that exact observation taken from the sensors (Step 2). This results in a more refined belief and the uncertainty is seen to decrease, this is shown by the green distribution.

Section 2.4.1 presented an overview containing the process and the mathematical formulations behind the Bayes Filter. However, there are vital considerations taken prior to arriving at the final formulation of the prediction and correction models. This includes the notorious assumption of a *complete system*, otherwise known as the *Markov Chain*. Prior to the presentation of the Bayes filter derivation process that arrives at Equation (2.14) and Equation (2.15), an overview of the theory concerning the Markov Chain is presented in the following section.

2.4.2 Markov Model

A stochastic process is deemed as a Markov Model (MM) if that process satisfies the Markovian property [21]. The Markovian property states that any estimate of the conditional probability distribution of a future state in the process X_{t+1}, given the present state X_t, depends only upon the present state and is conditionally independent of any

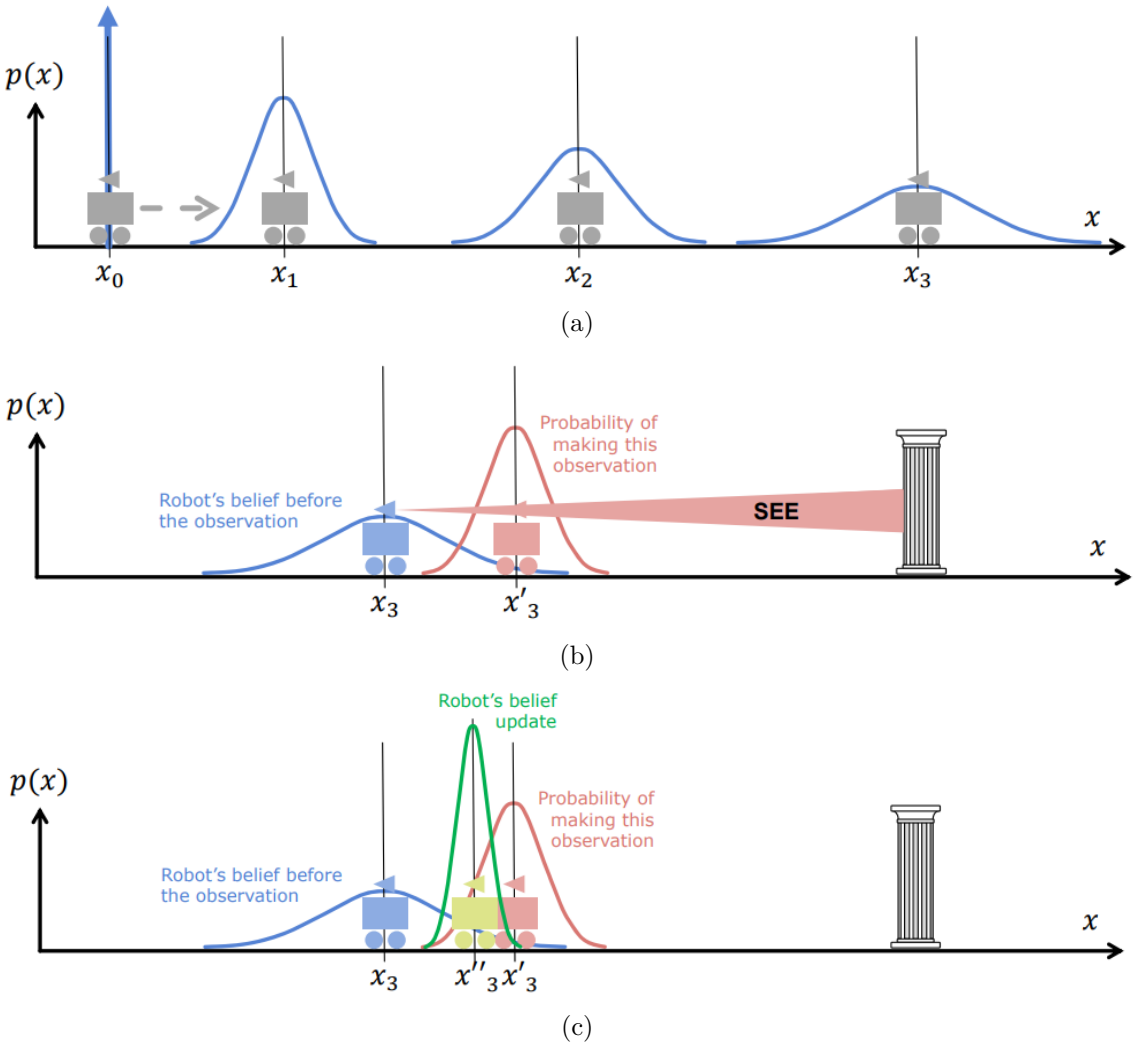


Figure 2.5: Belief propagation

previous states within the process, $X_{0:t-1}$ [22]. A graphical representation of a discrete MM is shown in Figure 2.6. This representation uses three integers as states that this example process could possibly adopt. Each strand represents a conditional probability of the state adopting the integer that the leading arrow is touching given the current state represented by the origin of the strand.

In the following section, the derivation of the Bayes Filter is presented. It also includes how the Bayes Filter assumes a Markov process as briefly introduced above and how this assumption influences the final derivation of the algorithm.

2.4.3 Mathematical Derivation of Bayes Filter

The overall aim of the Bayes filter algorithm is to calculate a posterior distribution from the corresponding posterior distribution one time step earlier. The following derivation is

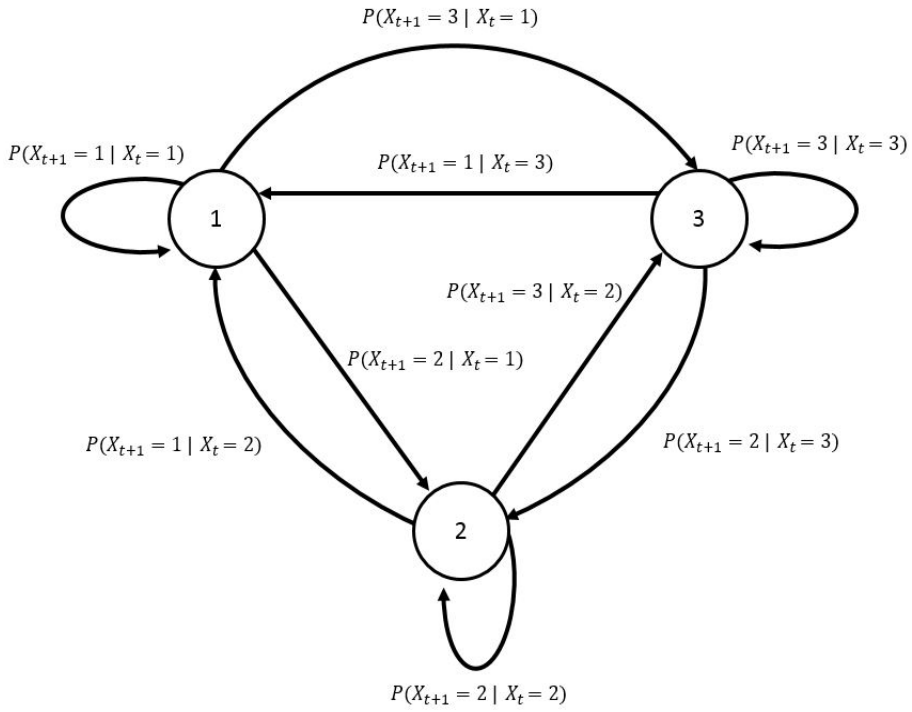


Figure 2.6: Markov Model

included to demonstrate how Bayesian inference, Bayes rule of conditional probabilities and MMs results in the formulation of the belief distributions previously shown in Equation (2.14) and Equation (2.15). The derivation of the Bayes filter starts with defining the Bayes rule of conditional probabilities as shown in Equation (2.16) and Equation (2.17), where x_t and u_t represent the state and the control input at time t, z represents external observations acquired from sensory systems and η represents the normalisation constant.

$$p\left(x_t \mid z_{1:t}, u_{1:t}\right) = \frac{p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t})}{p(z_t \mid z_{1:t-1}, u_{1:t})} \tag{2.16}$$

$$p\left(x_t \mid z_{1:t}, u_{1:t}\right) = \eta\, p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t}) \tag{2.17}$$

The assumption that is made in the design of the Bayes Filter is that it satisfies the Markov property following that presented in Section 2.4.2. Therefore, given that the current state x_t is known, the evolution of any future states are independent of past states, control inputs or observations. These variables are assumed to have no influence on the evolution of future states. This also holds when predicting the observations z_t given that x_t is known, therefore meaning that no past observations or control inputs provide additional information. This is represented in Equation (2.18) [8, 12].

$$p(z_t \mid x_t, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t) \tag{2.18}$$

Therefore, Equation (2.17) can then be represented as that shown in Equation (2.19) and further represented as Equation (2.20), this shows how the correction stage of the Bayes Filter located in Equation (2.15) is acquired.

$$p\left(x_t \mid z_{1:t}, u_{1:t}\right) = \eta\, p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1}, u_{1:t}) \tag{2.19}$$

$$Bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{Bel}(x_t) \tag{2.20}$$

The prior belief $\overline{Bel}(x_t)$ used in Equation (2.20) to calculate the posterior belief $Bel(x_t)$ is then expanded upon. Its initial representation is described in Equation (2.21).

$$\overline{Bel}(x_t) = p(x_t \mid z_{1:t-1}, u_{1:t}) \tag{2.21}$$

Then using the theorem of total probability located in Equation (2.10), Equation (2.21) is then written as shown in Equation (2.22).

$$p(x_t \mid z_{1:t-1}, u_{1:t}) = \int p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t})\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t})\, d_{x_{t-1}} \tag{2.22}$$

Once again, the Markov assumption is implemented to arrive at Equation (2.23). This implies that given knowledge of x_{t-1}, all past observations and control inputs provide no additional information concerning the evolution of x_t.

$$p(x_t \mid x_{t-1},\ z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t) \tag{2.23}$$

It is also noted that the posterior belief one time step earlier $Bel(x_{t-1})$ is represented as that shown in Equation (2.24), therefore Equation (2.21) can be simplified as shown in Equation (2.25).

$$Bel(x_{t-1}) = p(x_{t-1} \mid z_{1:t-1}, u_{1:t}) \tag{2.24}$$

$$\overline{Bel}(x_t) = \int p(x_t \mid x_{t-1}, u_t)\, Bel(x_{t-1})\, dx_t \tag{2.25}$$

Therefore, Equation (2.25) presents the prediction stage of the Bayes filter as described in Equation (2.14), further demonstrating how the implementation of Markovian assumptions and conditional probability theory results in the derivation of the Bayes filter algorithm. Two important aspects of the derivation process are related to Equation (2.18) and Equation (2.23). The formulations $p(z_t \mid x_t)$ in Equation (2.18) and $p(x_t \mid x_{t-1}, u_t)$ in Equation (2.23) represent what is known as the measurement probability and the state transition probability respectively [8, 12]. These probability densities are vital in the explanation of Gaussian based filters that are presented in the following section.

To summarise Section 2.4, the Bayes filter was presented and derived to demonstrate how the calculation of future state belief distributions concerning a robotic system is achieved. The underlying reason for the presentation of the Bayes filter is due to its relation with the algorithm known as the Kalman Filter (KF). A KF can be defined as the most relevant Bayes filter variant in practice. The main difference between the Bayes filter and the KF is that the KF assumes that the belief distributions included throughout Section 2.4 are of a multivariate Gaussian nature. Therefore, the following section includes the presentation of the theory behind the KF algorithm.

2.5 Gaussian Filters

Section 2.5 presents theory concerning Gaussian filters in the form of a linear KF and its non-linear variant known as an Extended Kalman Filter (EKF). It is to be noted that the prior introduction to the Bayes filter is important as the linear KF, which is introduced in Algorithm 1, is identical to the Bayes Filter if the system is of a linear Gaussian nature. However, the linear KF is optimal only for linear systems, therefore, the EKF is presented in Section 2.5.2, as in mobile robotics, systems are rarely identified as linear due to kinematic reasons [23].

2.5.1 Kalman Filter

The KF is as a linear-quadratic estimation (LQE) process which computes beliefs concerning states of a continuous nature. The KF employs a state predict-correct system identical to that of the Bayes filter, however, as previously mentioned the KF represents the belief distributions of such states in certain instances of time as Gaussians. Such Gaussian

distributions within the KF are multivariate and are defined as shown in Equation (2.26). These multivariate Gaussian distributions are modelled using a moments representation, where the moments of the Gaussian at time t are the mean μ_t and covariance Σ_t of the distribution [24].

$$p(x) = det(2\pi\Sigma)^{0.5}\ e^{-0.5\ (x-\mu)^T\ \Sigma^{-1}(x-\mu)} \tag{2.26}$$

The following two paragraphs present the formulations for the predict and correct stages of the KF. It is to note that all input and output values are accompanied with an caret symbol ($\widehat{\ }$). This denotes that this variable is an estimation therefore showing that the KF is an estimation process. Values accompanied with a bar symbol ($\bar{\ }$) denote that this is a prior estimation before the incorporation of the sensor observations .

2.5.1.1 Prediction

The definition of the next state transition probability distribution, $p(x_t \mid x_{t-1}, u_t)$ presented in Equation (2.23) is used to define the prediction stage of the KF. In order for the KF to be optimal, the system is required to be of a linear nature with added Gaussian noise [25]. Therefore, the next state probability is expressed as that shown in Equation (2.27). It is also important to note that the KF also assumes that the process is of a Markov nature.

$$\widehat{X}_t^- = A_t\ \widehat{X}_{t-1} + B_t\ u_t + \nu_t \tag{2.27}$$

Where, A and B are matrices described in a following paragraph and ν is a random Gaussian distribution representing an added system noise characterised by a zero mean and Q covariance shown in Equation (2.28).

$$\nu = \mathcal{N}(\mu_\nu, \Sigma_\nu) = (0, Q) \tag{2.28}$$

During the process of the KF, the state vector is the vector which holds all required states concerning the system that are to be estimated, the state vector is represented by X_t at time t. X_t is a column vector of m dimensionality as shown in Equation (2.29), where m is equivalent to the number of states that are required to be estimated during the process. For example, if a system was linearly constrained to traverse in a single direction, the states of the system may be modelled as $X_t = [s\ v\ a]^T$, where s, v and a may represent

system displacement, velocity and acceleration at time t, therefore this would signify a three-dimensional state vector. Each state component encapsulated within X_t is represented by a belief distribution modelled as a Gaussian, which is characterised by its best available estimate in that certain time instance. This best available estimate is the mean value of the Gaussian. Therefore, the state vector X_t is effectively a vector containing mean variables μ [26]. This is shown in Equation (2.30). The control input vector u_t at time t is the vector which holds all inputs into the system acquired from proprioception. The dimensionality of u_t is the number of control inputs that are utilised. This is shown in Equation (2.31).

$$X_t = [x_{1_t}\ x_{2_t}\ \dots\ x_{m_t}]^T \tag{2.29}$$

$$X_t = [x_{1_t}\ x_{2_t}\ \dots\ x_{m_t}]^T = [\mu_{1_t}\ \mu_{2_t}\ \dots\ \mu_{m_t}]^T \tag{2.30}$$

$$u_t = [u_{1_t}\ u_{2_t}\ \dots\ u_{n_t}]^T \tag{2.31}$$

Values A and B presented in Equation (2.27) are matrices utilised in predicting the estimation of the future state x_t. A is defined as a matrix that describes how the state temporally transitions in the interval *t-1* to t and is a square matrix of dimensionality $m \times m$. B is defined as the matrix that describes how the control input u_t changes the state from *t-1* to t, the dimensionality of B is $n \times l$.

2.5.1.2 Correction

The correction stage of the KF concerns the measurement probability $p(z_t \mid x_t)$ that was previously presented in Equation (2.18). Similar to the process within the prediction stage, the measurement probability must be linear in its arguments for the KF to be applicable. For the correction stage of the KF, the measurement probability is represented as that shown in Equation (2.32).

$$\widehat{Z}_t = C_t \widehat{X}_t^- + \upsilon_t \tag{2.32}$$

Where υ_t is a Gaussian measurement noise also characterisation by a zero mean and R covariance as shown in Equation (2.33).

$$\upsilon = \mathcal{N}(\mu_\upsilon, \Sigma_\upsilon) = (0, R) \tag{2.33}$$

The variable C located within Equation (2.32) is known as the observation matrix and its function is to map the state variables within X_t to the same structure to the observations acquired from the sensors. This enables the prediction of a set of observations based on the prior state $\widehat{X}_t^-$ acquired from the state transition prediction stage in Equation (2.27). The reason for the prediction of a set of observations is to acquire what is known as the residual denoted as Y. This is calculated by finding the difference between the predicted set of observations and the actual set of observations received by external sensory units Z_t. This is shown in Equation (2.34).

$$Y_t = Z_t - \widehat{Z}_t \tag{2.34}$$

The main function of the residual within the KF algorithm is to provide knowledge concerning how the prediction differs from observations from on-board sensors. The vital application of Y during the process of the algorithm is its interaction with the Kalman gain variable, K. The Kalman gain is defined as a relative weighting factor which enables the output of the filter to trust the observation (higher K value) or to trust more the prediction (lower K value). The calculation of the Kalman gain weighting factor is shown in Equation (2.35).

$$K_t = \widehat{\Sigma}_t^- \; C_t^T \; (C_t \; \widehat{\Sigma}_t^- \; C_t^T \; + \; R_t)^{-1} \tag{2.35}$$

Therefore, the presentation concerning the process of the KF algorithm as a whole is displayed in Algorithm 1.

To summarise the presentation of Section 2.5.1, the KF is a variant of the Bayes filter which represents beliefs as multivariate Gaussian distributions. The KF is optimal for linear systems which hold both linear motion and measurement models. In order for the KF to hold, the belief distributions are required to be modelled as Gaussian. Therefore, the updates within the prediction and correction stages see the Gaussian distribution undergoing transformations through linear functions. The result from this is best represented by Figure 2.7, where the passing of a Gaussian through a linear function

Algorithm 1 - Kalman Filter

Inputs $(\widehat{X}_{t-1}, \widehat{\Sigma}_t, u_t, Z_t, Q_t, R_t)$

Outputs $(\widehat{X}_t, \widehat{\Sigma}_t)$

Prediction

1: $\widehat{X}_t^- = A\widehat{X}_{t-1} + Bu_t$

2: $\widehat{\Sigma}_t^- = A_t\widehat{\Sigma}_{t-1}A_t^T + R_t$

Correction

3: $\widehat{Z}_t = C_t\ \widehat{X}_t^-$

4: $K_t = \widehat{\Sigma}_t^-\ C_t^T\ (C_t\ \widehat{\Sigma}_t^-\ C_t^T\ +\ R_t)^{-1}$

5: $Y_t = Z_t - \widehat{Z}_t$

6: $\widehat{X}_t = \widehat{X}_t^- + (K_tY_t)$

7: $\widehat{\Sigma}_t = (I - K_tC_t)\widehat{\Sigma}_t^-$

retains a Gaussian [8]. This is the case as long as the transformation function is linear. However, it is very rare that dynamic systems within the field of probabilistic mobile robotics are of a linear nature. Therefore, the inability to model non-linear systems using linear state transition and measurement models renders the use of standard KFs unsuitable. The following section presents a method for overcoming system non-linearity in the form of the Extended Kalman Filter (EKF).

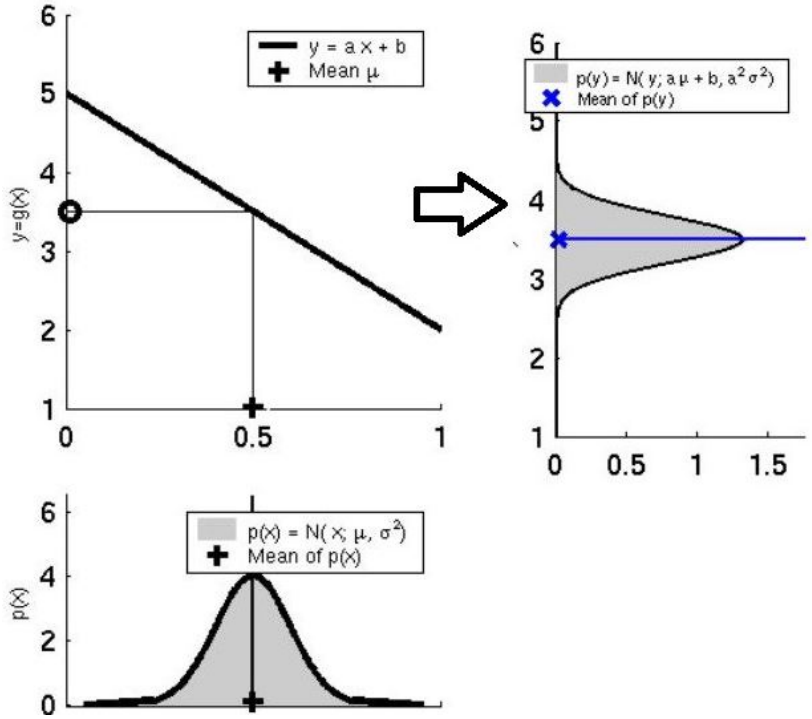


Figure 2.7: Transformation of a Gaussian through a linear function [27]

2.5.2 Extended Kalman Filter

In mobile robotics the assumptions of a linear system are rarely fulfilled in practice as all mobile systems have a degree of non-linear behaviour. For example, the constant translational and rotational velocity to execute a circular trajectory for a unidirectional system cannot be characterised as a linear model [28]. This is where the EKF differs from the standard KF as it is intended to estimate random variables of a non-linear system also using external observations from sources such as sensor feedback. The EKF is formulated in a similar nature to the standard KF, however, the EKF seeks to solve the non-linearities of a system through the linearisation approximation of the state transition and measurement models. This is done through the application of a first-order Taylor expansion represented as Jacobian matrices. In the formulation of the EKF, the state transition and measurement matrices (A and B) applicable within the KF are now governed by non-linear functions f and h shown in Equation (2.36) and Equation (2.37), which are the generalised representation of the prediction and correction stages respectively.

$$\widehat{X}_t^- = f(\widehat{X}_{t-1}, u_t) + \nu_t \tag{2.36}$$

$$\widehat{Z}_t = h(\widehat{X}_t^-) + \upsilon_t \tag{2.37}$$

The generalised format of the EKF algorithm is shown in Algorithm 2. An overview of the information flow during the EKF process is also shown in Figure 2.8. It is to be noted here that the EKF algorithm is an extended version of the KF, therefore, it has additional necessary steps executed in order to approximate the linearisation of the system. This linearisation is an approximation and is handled through three Jacobian matrices G_x, G_u and H_x which represent the state transition, control input and observation Jacobians respectively. The application of Jacobian matrices is carried out through a first order Taylor expansion process, which is covered in the following section along with the general formulation of G_x, G_u and H_x.

2.5.3 Taylor Expansion for Linearisation Approximation

As seen in the presentation of the KF for linear system models, mapping a Gaussian distribution through a linear function maintains the Gaussian nature of the distribution.

Algorithm 2 - Extended Kalman Filter

Inputs $(\widehat{X}_{t-1}, \widehat{\Sigma}_t, u_t, Z_t, Q_t, R_t)$

Outputs $(\widehat{X}_t, \widehat{\Sigma}_t)$

Prediction

1: $\widehat{X}_t^- = f(\widehat{X}_{t-1}, u_t)$

2: $G_{x_t} = \frac{\partial f}{\partial x}(\widehat{X}_{t-1}, u_t)$

3: $G_{u_t} = \frac{\partial f}{\partial u}(\widehat{X}_{t-1}, u_t)$

4: $\widehat{\Sigma}_t^- = G_{x_k}\widehat{\Sigma}_{t-1}G_{x_k}^T + G_{u_k} Q\, G_{u_k}^T$

Correction

5: $\widehat{Z}_t = h(\widehat{X}_t^-)$

6: $H_{z_t} = \frac{\partial h}{\partial x}(\widehat{X}_t^-)$

7: $S_t = H_{z_z}\widehat{\Sigma}_t^- H_{z_z}^T + R_t$

8: $K_t = \widehat{\Sigma}_t^-\ H_{z_t}^T\, S_t$

9: $Y_t = Z_t - \widehat{Z}_t$

10: $\widehat{X}_t = \widehat{X}_t^- + (K_t Y_t)$

11: $\widehat{\Sigma}_t = (I - K_t H_{z_t})\widehat{\Sigma}_t^-$

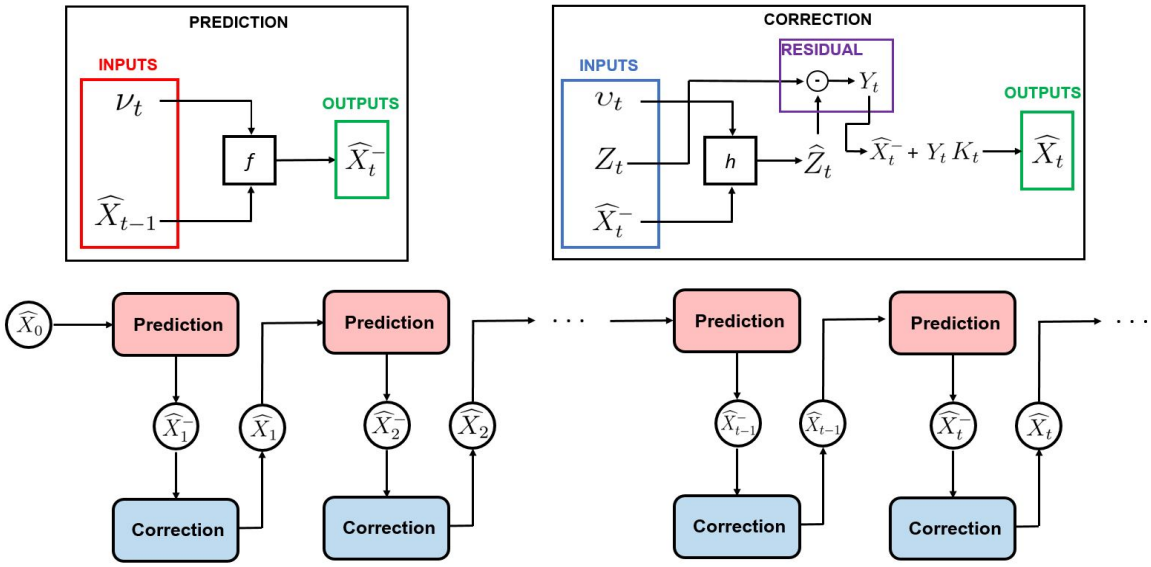


Figure 2.8: Overview of information flow in an EKF

However, when concerning non-linear models for systems with a degree of non-linearity, the requirement to propagate the Gaussian to estimate a future state means passing the distribution through the non-linear functions f and h. This becomes problematic as passing a Gaussian distribution through a non-linear function does not yield a Gaussian output as seen in Figure 2.9. This is where the first order Taylor expansion is employed [27]. The first order Taylor expansion within the EKF process seeks to approximate the linearisation

of the non-linear motion and observation models f and h. Prior to the presentation of the mathematics behind a first order Taylor expansion and the formulation of Jacobian matrices G_x, G_u and H_x used in the EKF, Figure 2.10 provides a visual representation of the aim that the linearisation approximation through the Taylor expansion seeks to achieve and the effects it has concerning the resulting distribution [29].

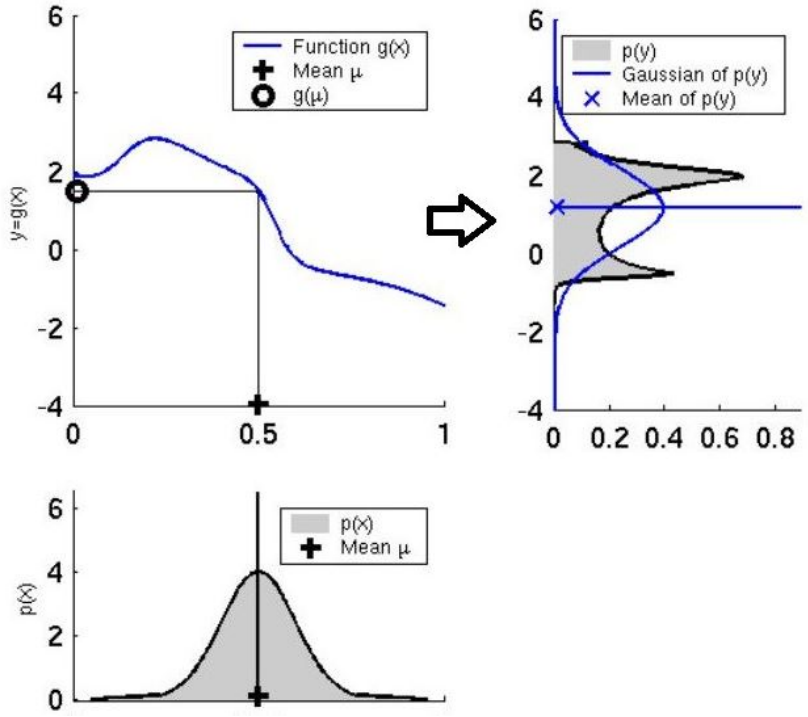


Figure 2.9: Transformation of a Gaussian through a non-linear function

The Taylor expansion approximates the linearisation at a set of specified points. This is shown in Figure 2.9, where the linear approximation of the non-linear function is modelled as a tangent line passing through the linearisation point. For example, consider a single dimension state vector $X = [\, x \,]^T$ with the non-linear motion model $f(\widehat{X}_{t-1}, u_t)$. The Taylor expansion will approximate f by taking the most likely value of the argument to use as the linearisation points. This most likely value for Gaussian distributions is the mean of the distribution, in the case of this example, this value is denoted as $\mu_{x_{t-1}}$. The formulation for this linearisation is shown in Equation (2.38) and its simplified variant shown in Equation (2.39).

$$f(\widehat{X}_{t-1}, u_t) \approx f(\widehat{\mu}_{t-1}, u_t) + \frac{\partial f}{\partial \widehat{X}_{t-1}}(\widehat{\mu}_{t-1}, u_t)(\widehat{X}_{t-1} - \widehat{\mu}_{t-1}) \tag{2.38}$$

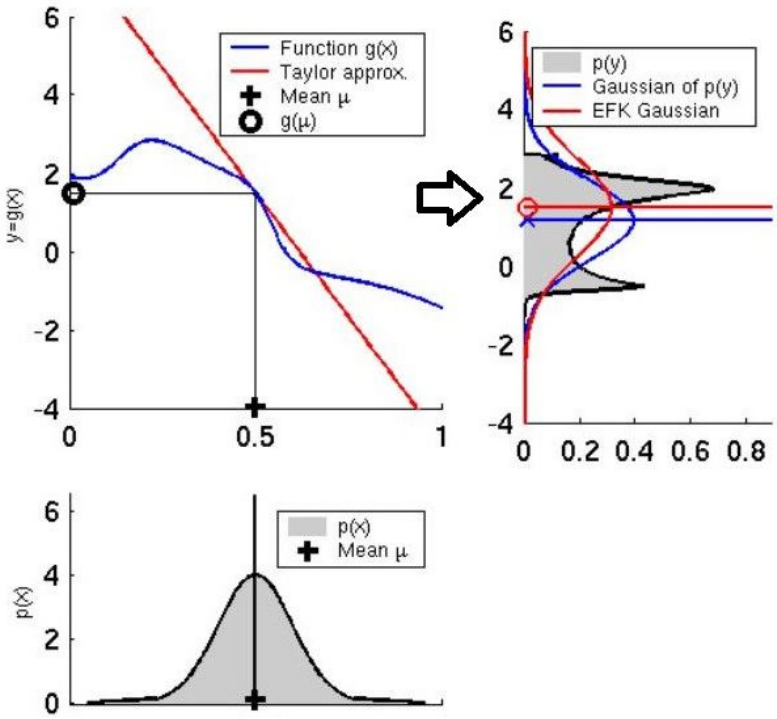


Figure 2.10: Taylor expansion linearisation approximation

$$f(\widehat{X}_{t-1}, u_t) \approx f(\widehat{\mu}_{t-1}, u_t) + G_x(\widehat{X}_{t-1} - \widehat{\mu}_{t-1}) \tag{2.39}$$

The important component within Equation (2.39) is the value of G_x which is considered as the state transition Jacobian matrix. This matrix is $n \times n$ in dimensionality where n is the number of estimated states. Therefore, following the same linearisation procedure, the three Jacobian matrices required for an EKF operation denoted as G_x, G_u and H_x for the state transition, control input and observation Jacobians respectively are shown in Equations (2.40) to (2.42). These Jacobians are dependent upon arguments that are non-constants so each Jacobian output will differ for each iteration of the filter.

$$G_x = \frac{\partial f}{\partial X}(\widehat{X}, u) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \vdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{2.40}$$

$$G_u = \frac{\partial f}{\partial u}(\widehat{X}, u) = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial u_1} & \vdots & \frac{\partial f_m}{\partial u_k} \end{bmatrix} \tag{2.41}$$

$$H_x = \frac{\partial h}{\partial X}(\widehat{X}^-) = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_i}{\partial x_1} & \vdots & \frac{\partial h_i}{\partial x_n} \end{bmatrix} \tag{2.42}$$

2.6 Conclusion

In conclusion, Chapter 2 covered the fundamental theories that are both studied and applied throughout this thesis. The following points summarise the content within Chapter 2.

- State estimation is a fundamental topic of probabilistic and statistical robotics.
- Modelling the kinematics of mobile and unmanned robotics platforms through initial system identification processes provides the ability to consistently monitor sensor output for anomaly detection and mitigation.
- Basic theories relating to joint and conditional probabilities and their relationship with Bayes' Rule of probability are what mark the foundations of probabilistic state estimators.
- Stochastic Gaussian filters such as the KF and the EKF are based on belief distribution estimation which stem from the Bayes Filter.
- The control related algorithms known as the KF and EKF are assumed to follow what is known as a Markov Model (MM), which states that the belief distribution of a state relating to the system at time t depends only on the belief distribution of that state at time *t-1*.

Chapter 3 which follows is titled *Robotic Systems Design*, where the robotic systems that were designed as data collection and environmental exploration tools for this thesis and for the DigiArt project itself are presented.

Part II

Unmanned System Design & Mathematical Modelling

Chapter 3

Robotic Systems Design

Chapter 3 provides an overview on the robotic systems that were designed for the deployment of experimental payloads, control algorithm implementation and general data collection procedures as a part of the DigiArt project. During this thesis, two robotic platforms were designed for two different applications which are described throughout this chapter. Both systems were designed using skid steered motion to enable high manoeuvrability and for its simple and robust mechanical structure, utilising more of the maximum system weight for sensor payloads and experimental equipment. However, unlike a differential drive system, in the case of the skid steered locomotion scheme it can be somewhat difficult to develop kinematic and dynamic models that can accurately describe this motion. Therefore, this chapter provides an insight into skid-steering and describes the approach undertaken to model the designed robotic systems used for this thesis.

3.1 Skid-Steer Kinematics

Skid-steering is a commonly deployed locomotion mechanism for both manned and unmanned mobile ground vehicles. In comparison to a steering scheme such as Ackerman, skid-steering has no steering mechanism and instead changes the attitude of the vehicle by adjusting the speed of the left and right side wheels or tracks. The zero-radius turn capabilities of a skid-steered system allows robust manoeuvrability on all terrains and operating within condensed and harsh environments. However, the dynamics of skid-steered systems are complex due to large amounts of track or wheel slippages when executing certain manoeuvres that require an alternation of the vehicle's attitude. Therefore, such occurrence of slippage results in the accumulation of time related drifting errors, further resulting in

the inability to accurately model this steering scheme using the control inputs provided into the system. The steering structures for tracked, four and six wheeled skid steering systems are shown in Figure 3.1 alongside a differential drive system. The underlying difference between an ideal differential drive system shown in Figure 3.1a and skid-steering systems is that the tread contact points with the terrain are limited to a single point. These points are represented by small markers on the wheels of the model shown Figure 3.1a. Tracked differential drive systems epitomise skid-steering due to the large contact points with the terrain shown in Figure 3.1b. Also, skid-steering in wheeled systems has several contact points that correspond to mechanically linked wheels shown in Figure 3.1c and Figure 3.1d.

Also, another underlying difference between differential drive and skid-steered systems is the locations of the instantaneous centres of rotation (ICRs) of the left and right treads and the centre of mass. An ICR is a fixed point that has zero velocity in relation to a rigid body undergoing planar movement. As shown in an initial and simplistic schematic in Figure 3.1, the ICRs for the differentially driven system are constant and coincide with the ground contact points of the wheels. However, the skid-steer system contrasts in terms of the ICRs being dynamically dependent and lying outside of the wheel centre lines. This offset is another result of wheel slippage experienced in a skid-steered motion scheme and ICR locations are influenced by the location of the robot's centre of mass as shown in Figure 3.2, where the ICR on the side where the centre of mass is located is seen to be closer to the robot body.

Figure 3.3 expands upon the kinematic scheme for a skid-steered (SS) robot. The schematic shows the positions of the ICRs for the centre of mass, left and right wheels as well as wheel velocity vectors as a skid-steered robot manoeuvres along an arc. The ICRs of the left and right wheels can be determined by taking a perpendicular line from the tail of the translational velocity vectors. The intersection point of those two perpendicular lines is where the ICRs (ICR_l, ICR_r) can be calculated. This is also the case for the ICR of the robot body ICR, which is calculated as the intersection point between the perpendicular line passing ICR_l, ICR_r and the robot's centre of mass. It is noted that ICR_l, ICR_r and ICR lie on a line parallel to the x-axis of the robot's body frame.

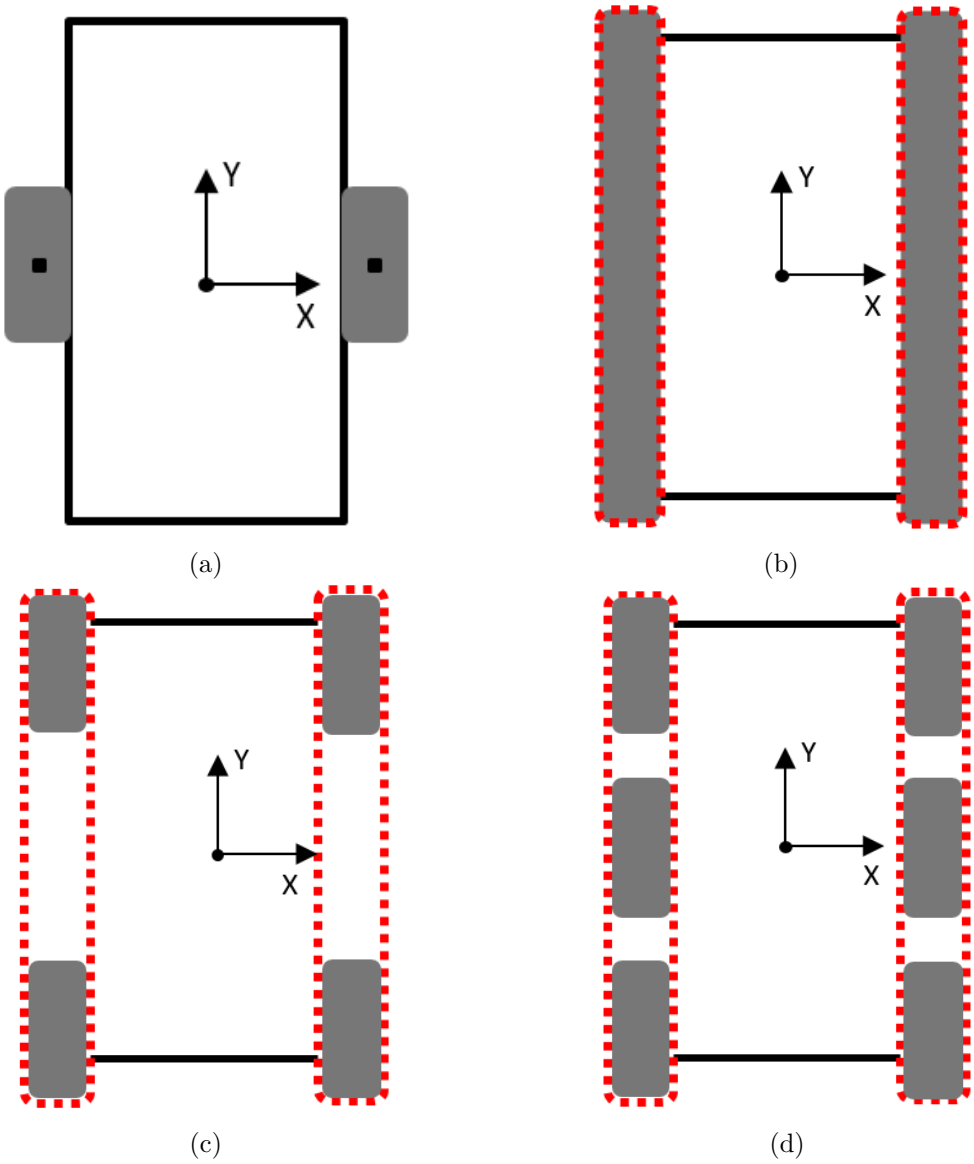


Figure 3.1: Left and right treads on the plane for (a) differential drive, (b) tracked system, (c) four-wheeled skid-steer system, (d) six-wheeled skid-steer system

3.2 Unmanned System Design

Section 3.2 presents an overview of the designed unmanned systems used for experiments throughout this thesis. As discussed, the justification as to why the SS schematic was utilised was due to its robustness, simplicity and ability to manoeuvre in confined areas. Each section describing the two unmanned platforms presents a component list providing details as to which payloads were mounted to each platform. Such information can be found in Section 3.2.1 and Section 3.2.2.

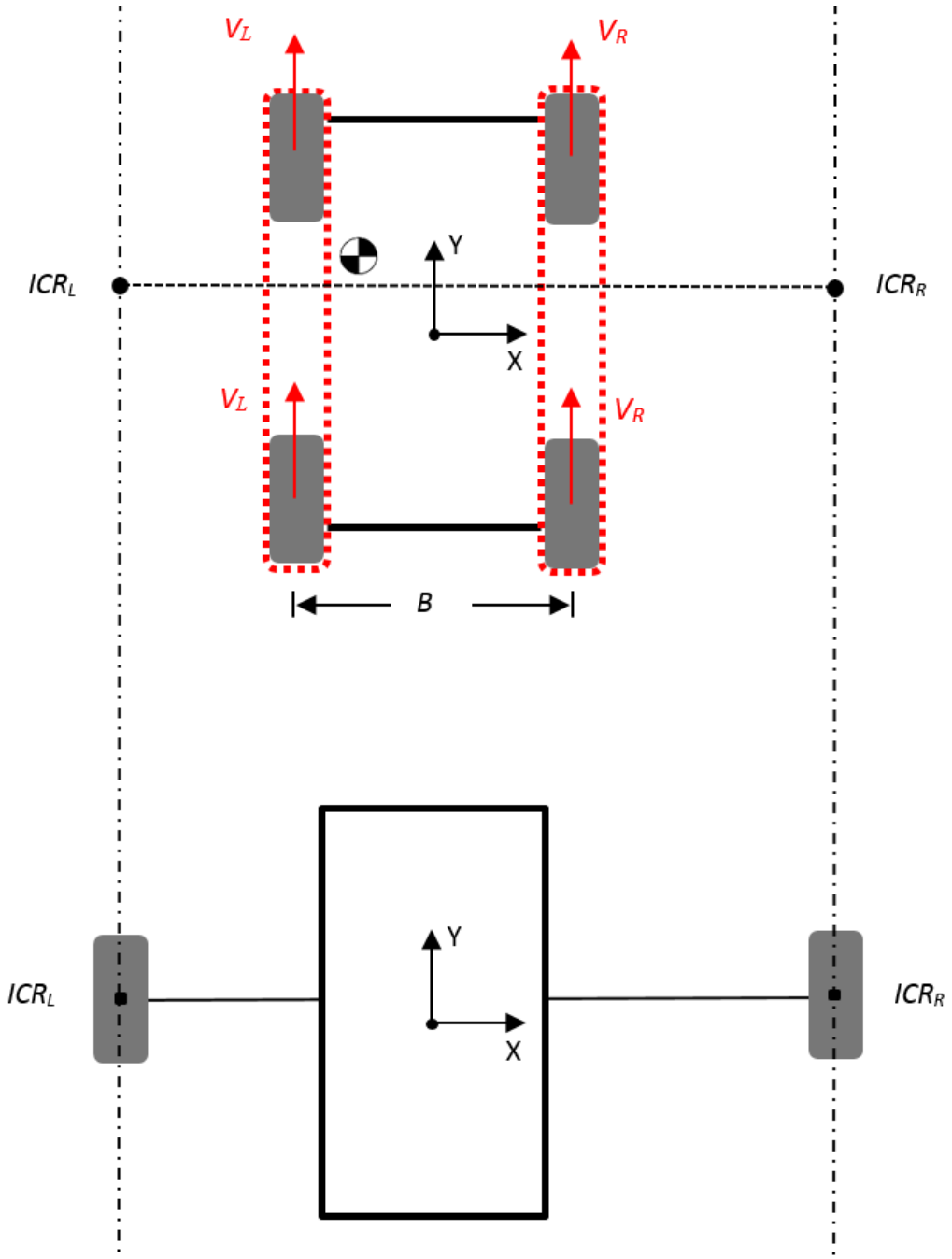


Figure 3.2: Kinematic equivalence between wheel ICRs of a skid-steered system and an ideal differential drive system

3.2.1 Platform 1

Platform 1 was designed as a robust and heavy-duty developmental system. For initial testing concerning the quantification of erroneous feedback from experimented positioning systems, Platform 1 was designed with capabilities to carry heavy sensory payloads to acquire data and compare the results from each payload in reference to a ground truth metric. Platform 1 was the system used for the work carried out in [31, 32]. The

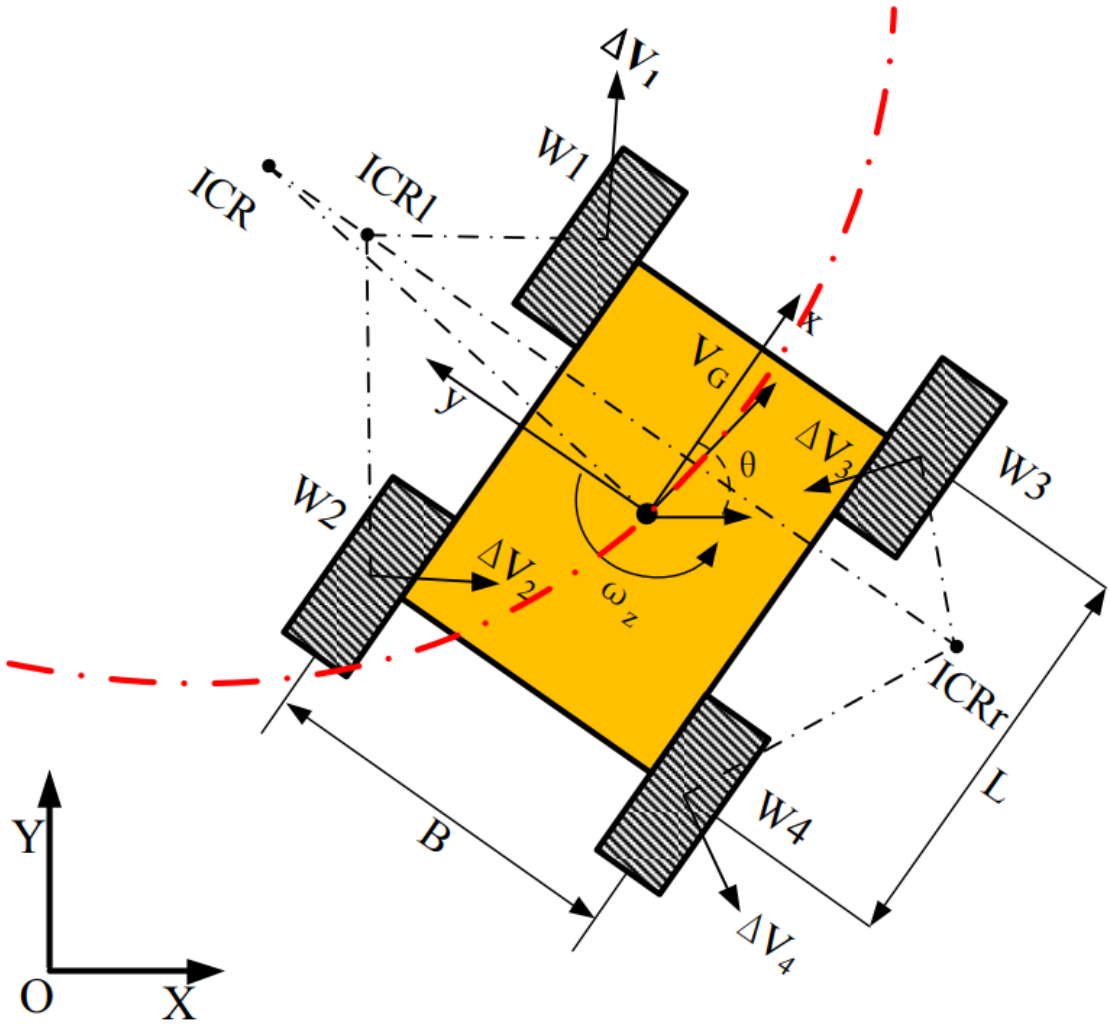


Figure 3.3: Kinematic schematic of the skid-steered locomotion scheme [30]

component list of this platform is shown in Table 3.1, where the information under the heading *Interface Type* is the method of interface between a sensor component and an on-board embedded system that is tasked with data logging. Additionally, Figure 3.4 provides an image of the final design of Platform 1.

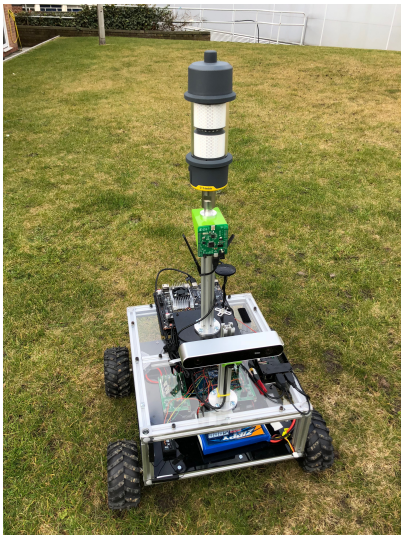
Table 3.1: Platform 1 Component List

Component	**Interface Type**	**Mass (g)**	**Function**
Raspberry Pi 3 (R-Pi)	N/A	42	The R-Pi is the embedded device which deploys the Kinetic version of the ROS architecture. The R-Pi is equipped with an ARM image of Ubuntu Mate and this device is utilised as the connection point for all sensor payloads apart from the Stereo Labs camera on Platform 1.
Jetson TX1 Development Board	N/A	1540	The Jetson TX1 is the embedded device which also deploys the Kinetic version of the ROS architecture. The TX1 is equipped with a desktop image of Ubuntu 16.04 Xenial Xerus. The TX1 has the capabilities to run the graphics drivers for the Stereo Labs camera.

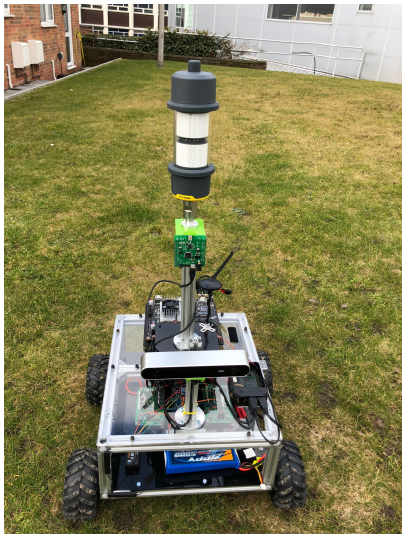
Velodyne HDL-32E	Ethernet	1300	Vision based system providing stereo visual odometry (SVO) to estimate robot pose
Trimble AT-360 Reflective Target	N/A	630 (with battery)	Acts as the interface between robot and RTS for active tracking and ground truth metric generation
Pixhawk FCU & GPS	USB Serial	38	Provided IMU capabilities as well as GPS output for robot position tracking. Also provided internally fused EKF output generated from native firmware algorithm.
Bosch IMU	i2c	3	Provided IMU capabilities in terms of estimated linear accelerations and both estimated linear and angular velocities.
Pozyx UWB Tag	USB Serial	12	UWB Positioning rover tag which communicates with the external static reference beacons
FrSky X8R Receiver	Pulse Width Modulation (PWM)	17	Receiver unit to enable remote operation
Quimat DC H-Bridge Motor Driver	N/A	82	Motor controller component

4 x DC Motor-25 GA 370 Standard Version	N/A	140	System propulsion
4 x Magnetic Encoders	UART	3	Provides feedback concerning angular velocities for all four wheels
Arduino MCU	USB Serial	37	Acted as the interface between the R-Pi and the magnetic encoders
3 x 3S 11.4V Zippy Lithium Polymer Battery	N/A	177	Power sources for propulsion, TX1 and R-Pi

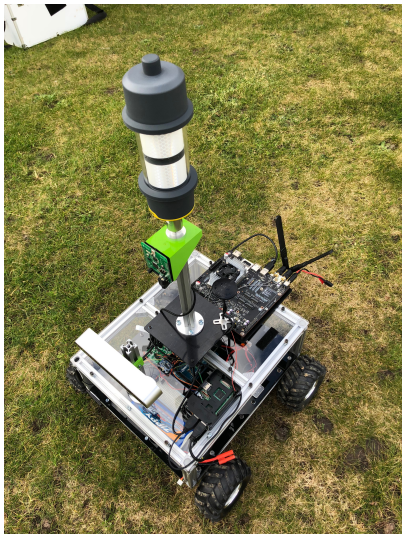
(a)



(b)



(c)



(d)

Figure 3.4: Platform 1

3.2.2 Platform 2

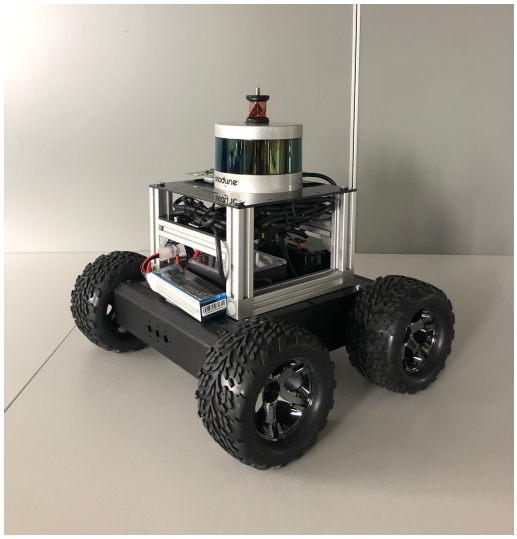
Platform 2 was designed as the final system that would be used to navigate through environments to generate 3D point cloud maps. Platform 2 is a lightweight and compact system which is easily deployable and transportable and was used for the work carried out in [33]. The component list of this platform is shown in Table 3.2, where the information under the heading *Interface Type* is the method of interface between a sensor component and one of the embedded systems that log its output, identical to that of Platform 1. Additionally, Figure 3.5 provides an image of the final design of Platform 2.

Table 3.2: Platform 2 Component List

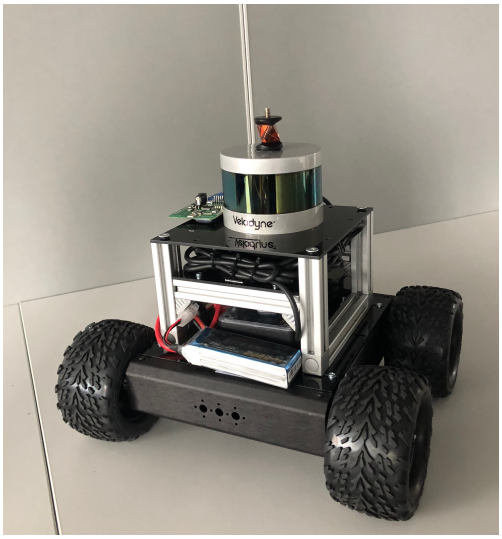
Component	**Interface Type**	**Weight (g)**	**Function**
Raspberry Pi 3 (R-Pi)	N/A	42	The R-Pi is the embedded device which deploys the Kinetic version of the Robot Operating System (ROS) architecture. The R-Pi is equipped with an ARM image of Ubuntu Mate and this device is utilised as the connection point for all other components on Platform 2
Leica Optical Mini-Prism	N/A	50	Acts as the interface between robot and RTS for active tracking and ground truth metric generation
Velodyne VLP-16 Puck	Ethernet	600	LiDAR system which collects 3D point cloud data
Pozyx UWB Tag	USB Serial	12	UWB Positioning rover tag which communicates with the external static reference beacons

Pixhawk FCU	USB Serial	38	FCU set to operate in ground-based mode. Provides data from inertial measurement unit (IMU). Also provides internally fused EKF output generated from the native firmware algorithm
FrSky X8R Receiver	PWM	17	Receiver to enable remote operation
Sabertooth 2x10 R/C Motor Driver	N/A	30	Motor controller device for pulse width modulation (PWM) handling from the XR8 receiver
2 x Magnetic Quadrature Encoders	UART	3	Provide updates concerning velocities for the left and right rear wheels. Connected to the Teensey MCU
Teensy 3.2 MCU	Serial	3	Acts as the interface between the quadrature encoders and the R-Pi
2 x 3S 11.4V Zippy Lithium Polymer Battery	N/A	177	Used to power the system and the LiDAR

4 x 12V DC 30:1 Gear Motors	N/A	160	System propulsion

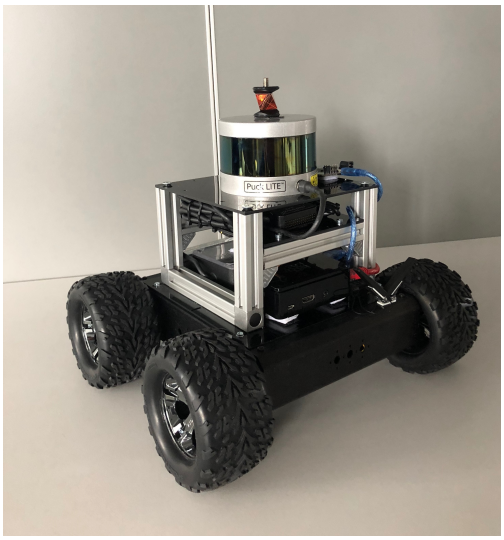


(a)



(b)

(c)



(d)

Figure 3.5: Platform 2

3.3 Conclusions

In conclusion, Chapter 3 firstly provided brief theory based on the kinematics of the skid-steered locomotion scheme. Chapter 3 then expanded into the presentation of information considering the designed robotic platforms used to conduct the experimental procedures executed in this thesis. The following summarises the main conclusions arising from the work presented in Chapter 3.

- The estimation of ICRs and amount of wheel slippage occurrence was accounted for as a component of the process noise when formulating the EKF.
- Platform 1 was an initial heavy duty developmental platform where many payloads were mounted for experimental testing,
- Platform 2 was the final compact design where the finished algorithms were implemented and deployed within the intended environment.

Chapter 4 which follows presents the initial kinematic modelling of the robotic systems presented in Chapter 3. This includes the motion modelling of the systems in a 2D nature and its integration with a wireless positioning technique. This is all encapsulated within a predict-correct state estimation algorithm to enable constant tracking of the robotic system within its operational environment.

Chapter 4

Two-Dimensional Localisation Problem

In terms of mobile unmanned systems and their relations to state estimation, an initial decision was made to firstly simplify the process. This included assessing the estimation of the states of an unmanned system as a 2D localisation problem. This includes estimating motion transition of each component of the 2D pose of the robot, which includes the Cartesian positions (x, y) and heading (ψ). In this chapter, the 2D pose will be defined as $\chi = [\, x\ y\ \psi\,]$. The ability for the robot to provide precise updates concerning the state of χ is highly dependent upon the positioning sensors on-board. However, it is notorious that measurements provided by sensors operating on-board robotic systems incorporate noise, which inherently limits what the sensor can perceive. The generation of sensor noise results in the occurrence of unpredictable perturbation effects on the feedback measurements. The existence of sensor measurement noise is a component of a notorious and key element of robotics: *Uncertainty*.

As discussed, the presence of sensor noise affects the measurement feedback that each sensor produces. However, the implementation of sensor fusion algorithms enables error compensation from the multiple sources of sensory information. It is a beneficial approach to fuse measurements acquired from multiple sources of information, as updates from multiple perceptions are considered more reliable in comparison to using singular measurement sources as a standalone. Therefore, an EKF localisation algorithm is implemented using a UWB positioning system fused with gyroscopic outputs and feedback from rigidly mounted quadrature wheel encoders. However, prior to the design of the EKF, the uncertainty of each measurement source is quantified through the implementation of various structured

approaches. It is for this reason that all tests conducted throughout this thesis are accompanied by a *ground truth metric*, which provides a source of information that is considered as the ideal and true reference. Thus, the remainder of Chapter 4 provides detailed insights into the error characterisation techniques and methodologies implemented to quantify the uncertainty of the positioning sensors used to tackle the 2D localisation problem. Chapter 4 then demonstrates how such metrics are then integrated into an R-EKF localisation algorithm for sensor fusion to produce improved position estimations on a 2D plane compared to traditionally implemented algorithms.

It is to be noted that an assessment of an unmanned system in 2D only covers half of the problem, as the system experiences motion in 3D when functioning in intended operating environments. However, Chapter 4 presents the foundational assessments in 2D, therefore in the formulation of the R-EKF, the state estimation of the 2D pose of the robot only applies. It is the following chapter in this thesis where the system is modelled and sensor fusion procedures are applied in 3D.

4.1 State of the Art

Section 4.1 presents the state of the art concerning 2D localisation for mobile unmanned systems. It assesses this topic by splitting the literature into four categories which are listed below.

- Dynamic Unmanned System 2D Localisation - Review of the current and previous state of the art begins with presenting the generalised topic of 2D localisation for mobile unmanned systems. It presents studies conducted by influential robotics institutions and various sensory implementations.

- Dynamic Unmanned System 2D Localisation & UWB - The topic is then narrowed down to the presentation of the state of the art concerning the integration of dynamic unmanned system 2D localisation applications and UWB.

- Robotic Total Station Tracking & Dynamic Unmanned Systems - State of the art technology and implementations for optical tracking of dynamic robotic systems using an RTS are then presented.

- Robotic Total Station Tracking & Dynamic Unmanned Systems & UWB - The topic of RTS integration with mobile robotics is then narrowed down further to present the state of the art concerning RTS tracking of dynamic unmanned systems that are functioning using UWB.

It is important to note that in terms of mobile robotics, the use of system identification for motion model generation and its integration with observations acquired from sensory systems is a critical and fundamental aspect of mobile robotics theory and implementation. Simply using a sensor alone without initial ideas of how the system is expected to temporally transition is considered as impractical. This is due to the system's inability to compare the sensor output with what the system's kinematic motion model provides in that certain instance. This therefore restricts the opportunity to assess whether the sensor is behaving irregularly and providing incorrect feedback.

4.1.1 Dynamic Unmanned System 2D Localisation

An early study into the field of 2D mobile robot localisation was conducted by *Leonard et al.*[34], which presents the development of an EKF within a known environment. Although this implementation is only successfully deployed if the environment is known prior, it demonstrates a framework for system identification through kinematic modelling for recursive robotic state estimation implementation. The algorithm designed uses an odometry motion model and demonstrates how this is utilised within an EKF fused with geometric beacon updates from a sonar system to estimate the 2D pose of a dynamic robotic system. The geometric beacons are acquired using an a priori map of the environment and without a known prior map may become a limitation to the design algorithm.

A study was conducted by *Kiriy et al.* [35] which presents the design of a three state EKF localisation algorithm for 2D pose estimation. The design of the EKF considers the motion model of the system and uses this model with sensor feedback from a fibre-optic gyroscope (FOG), wheel encoders and cameras operating in conjunction with visual odometry (VO) algorithms for landmark identification. Although *Kiriy et al.* had access to higher performance equipment in FOGs and deployed different sensory systems compared to that used in this Chapter, the interesting component of the study conducted by *Kiriy et al.* is related to the system identification process and shares a similar identification process to

the work presented in this chapter of the thesis. The major findings of the aforementioned study were that the EKF localisation algorithm achieved an average Cartesian error of 0.352m with a standard deviation of error at 0.296m and a heading error of 0.244° with a standard deviation of 3.488°. The authors suggest that in order to optimise their filter, an online assessment of the noise covariance matrices within the EKF should be taken upon each iteration of the filter to provide more of a comparison between the output of the motion model and the output of the measurement model.

Thrun et al. [36] conducted an early study focusing on the Monte Carlo Localisation (MCL) technique for mobile robotics. MCL techniques represent the belief of a robot's pose through a number of weighted hypotheses (samples), which seek to estimate the posterior state under a Bayesian framework. MCL through sample generation is the traditional method used for belief estimation with the high performance yet highly computationally inefficient particle filter. The study presented by *Thrun et al.* concludes that the MCL approach that was developed yielded better localisation performance in comparison to previous implemented and traditional techniques. The underlying reason for presenting this study is that it seeks to identify *Thrun et al.* as being one of the leading researchers in this field.

4.1.2 Dynamic Unmanned System 2D Localisation & UWB

Section 4.1.2 presents the state of the art in system design and implementation concerning mobile robot 2D positioning and localisation using UWB.

A study conducted by *Conceiçã et al.* [37] deploys a positioning technique using UWB as a cost effective and alternative solution to GNSS for agricultural robots. This study assesses a low cost and robust UWB system known as Pozyx, which is a commercial Time-of-Arrival (TOA) based UWB positioning system. *Conceiçã et al.* firstly statically characterise the mean and standard deviation of the estimation error produced by the Pozyx network. This was achieved by increasing the true distance between two UWB devices and measuring the resulting error. *Conceiçã et al.* then integrate this error variance into an EKF as the observation covariance value R. The motion model for the designed EKF is based on the odometry motion model for 2D pose estimation. The major findings of this study concluded by *Conceiçã et al.* state that the designed EKF produces a trajectory that has a reduction in noise and fewer outliers in comparison to the Pozyx native algorithm. The results

concerning localisation estimation performance of the EKF are not quantified, however, a graphical representation is provided which uses an open source Simultaneous Localisation and Mapping (SLAM) algorithm for the ground truth metric.

Wang et al. [38] presented a paper based on the development of a fast localisation and mapping algorithm based on UWB input and utilisation of visual SLAM. The authors' reasoning behind the utilisation of UWB was down to its ability to be robust when it comes to multi-path scenarios of effects caused by non-clear-line-of-sight (NCLOS) conditions [39] [40]. The main finding of this study states that the fusion of UWB with the traditional SLAM implementation yields more precise trajectory generation in comparison to the visual SLAM technique alone. *Wang et al.* then conclude by stating that the incorporation of the two measurement sources with multiple constraints, as well as its combination with a novel graph optimisation technique, enables the generation of drift-free localisation for mobile unmanned systems.

Pong et al. [41] conducted a study that consisted of the development of an EKF sensor fusion algorithm using UWB and 2D LiDAR for 2D SLAM. *Pong et al.* stated that the necessity for such fusion of data may present a more optimal methodological approach to range-based SLAM applications, which has also been studied in [42] and through a more probabilistic approach presented in [43]. This study is similar to that presented by *Wang et al.*, however, the deployed SLAM technique differs from visual to optical based SLAM, where the two techniques deploy vision and laser sensory units respectively. *Pong et al.'s* reasoning behind this approach suggests that the integration of the nature of the two measurement sources can complement the errors of each, where they suggest that the UWB aids in the reduction of LiDAR related drift and LiDAR aids in the improvement of the UWB position estimations. This study demonstrates that, with correct system identification, modelling and manipulation of data, two fairly contrasting methods of environment perception sources can achieve the design of a fully functioning and effective 2D localisation system for a ground-based mobile robot.

4.1.3 Robotic Total Station Tracking & Dynamic Unmanned System 2D Localisation

Section 4.1.3 focuses on the aspects of state of the art in system design and implementation concerning the utilisation of an RTS for optically tracking an unmanned system for ground

truth metric generation.

A study conducted by *Martinez et al.* [44] presents the novel design of an intensity-difference VO algorithm for unmanned ground-based planetary rovers. *Martinez et al.* state that such a designed algorithm only functions in sunlit conditions. Within this study, a Trimble S3 RTS was utilised as a means to produce a ground truth metric for algorithm validation. The RTS was taken to be the optimal method for positioning and was incorporated within the final stage of the experimental procedures for algorithm validation only. The findings of this study yield a 0.9% position error generated by the VO algorithm in comparison to the RTS. Although this study uses an RTS to remotely track an unmanned system through the use of a miniature optical prism similar to the work presented in this Chapter 4, the RTS is only utilised for final algorithm validation. The work presented in Chapter 4 incorporating the RTS with the UWB sees the RTS being used as a main component with the framework constructed to oversee the design of an EKF state estimation algorithm. The RTS is used from start to finish to quantify the uncertainty of the UWB system for certain parameters within the EKF, and is then used for final algorithm validation identical to the study conducted by *Martinez et al.*.

Although a ground-based unmanned system is considered in this thesis, there are plans mentioned towards the end of this thesis concerning transferring and adapting the work conducted and the systems designed, to operate on UAVs. Therefore, a few studies concerning optical tracking of UAVs using an RTS are considered. A study conducted by *Roberts et al.* [45] investigates the capabilities of an RTS to track a small scale unmanned aerial system (UAS). *Roberts et al.* state that the motivation of this study was to produce more improved geo-referenced aerially captured images in comparison to those provided by standard navigational GNSS. The conclusions gathered as a result of this study state that it was possible to obtain decimetre levels of geo-referencing accuracy using the RTS in comparison to the accuracy of standard navigation GNSS at 1-10m. *Roberts et al.* also state that this result can be improved upon with better time synchronisation between the RTS and UAV. A second study conducted in the area of UAV tracking using an RTS is one carried out by *Maxim et al.* [46] which focuses on UAV guidance using an RTS for fabrication site assessment. In the case of this study, an RTS is not utilised as a means to validate a system or a designed algorithm, it is used as input in order to control the UAV. The position estimations acquired by the RTS are sent to the on-board commercial open

source FCU known as the Pixhawk, which deploys the PX4 Autopilot, ROS. ROS functions through a message publisher-subscriber protocol, where sub-systems known as nodes can request permissions to subscribe to data that is being transferred over the network. The major findings concluded by *Maxim et al.* state that the use of an RTS for UAV guidance is first of all possible and effective and secondly, can be used as a large site exploration method for architectural structures.

4.1.4 Robotic Total Station Tracking, Dynamic Unmanned System 2D Localisation & UWB

Section 4.1.4 presents the state of the art concerning the integration of optical RTS tracking of unmanned systems that are operating through UWB. This is a more specific assessment to that presented in Section 4.1.3.

A stand-out study conducted by *Dobrev et al.* deploys an RTS to optically track an unmanned ground vehicle that is equipped with a UWB transceiver node for positioning. *Dobrev et al.* uses an EKF to fuse data from proprioceptive sensors, UWB and ultrasound which is used to update the heading of the robot. Each source of information has its own measurement model and the EKF is considered as a multivariate EKF, where the filter is updated in two separate stages. The design and implementation of the algorithm is then compared to a reference trajectory produced by a Leica RTS. However, in this case, the RTS was not used to quantify the range error of the UWB device as it is done for the work in Chapter 4, additionally, the study conducted by *Dobrev et al.* was published at a later date than the work presented in this Chapter, which can be found in [32].

4.2 Ultra wide-band Positioning

Section 4.2 presents an overview of UWB technology and the fundamentals of the UWB communication technique. It provides an in-depth analysis of UWB positioning techniques, which is presented in Section 4.2.2.

4.2.1 Ultra wide-band Technology Overview

UWB is a wireless communication technique that transmits and receives narrow radio pulses at a nanosecond rate. Unlike conventional radio frequency identification (RFID) based systems that operate on single bands, UWB transmits over a broad spectrum of radio

frequencies [47]. The ability to transmit over this extensive bandwidth decreases the power spectral density, therefore enabling UWB based systems to avoid interfaces with other radio frequency (RF) signals [48, 49]. Furthermore, as a result of the large operating bandwidth, the UWB system obtains a high time resolution; therefore, positioning techniques that deploy time techniques can provide accurate range estimations between transceiving nodes [50]. For the purpose of the experimental procedures conducted in Chapter 4, a low cost and lightweight integrated UWB sensor network system known as Pozyx was implemented [51]. This small scale system functioned through intercommunication UWB nodes categorised into static beacons (reference points) and mobile rover tags. An example image of the system can be seen in Figure 4.1.

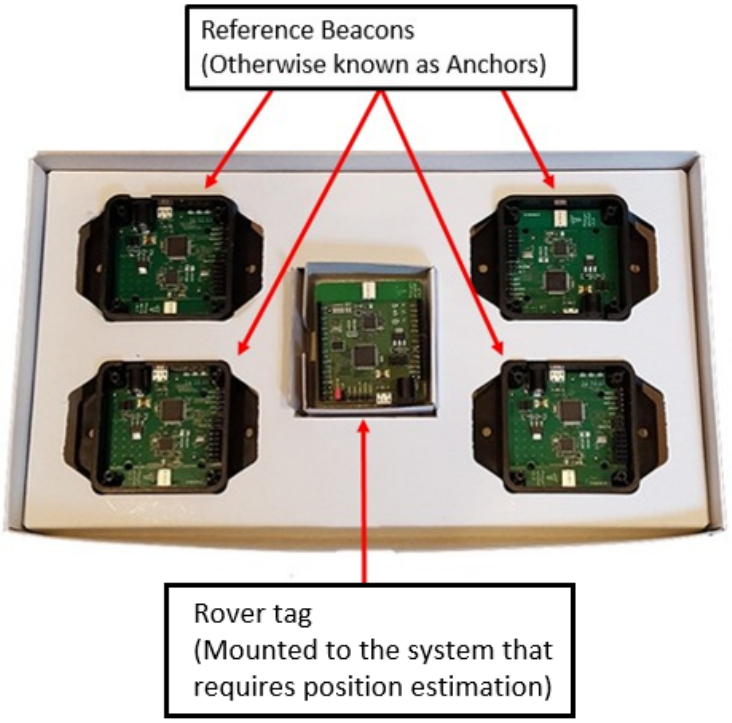


Figure 4.1: Pozyx UWB sensor network [51]

The Pozyx system can be used for robotic and non-robotic applications within research and other industrial institutions. This system is beneficial in that area as it deploys its own native positioning algorithm through tri-lateration and multi-lateration positioning techniques. This makes the Pozyx a 'plug and play' out of the box system. This characteristic of the Pozyx is the main reason why this specific system was utilised as it provided the opportunity to demonstrate the development of an EKF and how this algorithm performs in relation to the native position algorithm deployed by the Pozyx. The

communication between each Pozyx transceiver is achieved through time of flight (TOF) technique, where each transceiver measures the time taken to receive a signal from another. This method enables the calculation of the distances between interlinking transceivers. Traditional positioning techniques for wireless systems and UWB specifically are presented in Section 4.2.2.

4.2.2 Ultra wide-band Positioning Techniques

Expanding upon the introduction to the open source Pozyx system in Section 4.2.1, Section 4.2.2 transitions into the presentation of UWB positioning and its relation to robotics. Section 4.2.2 discusses how the UWB communication technique is utilised to provide updates concerning the location of an unmanned system within its operating environment. In the case of UWB and position estimation of a mobile platform, a set of static reference beacons are used to determine the location of a rover node within the environment enclosed by the beacons [49]. An enhanced understanding of this configuration can be seen in Figure 4.2. However, estimations of the position of a rover node acquired from a set of reference beacons have a degree of estimation error encapsulated within. This measurement error therefore results in an uncertainty region enclosed by the position estimations, as shown by the shaded blue region in Figure 4.2. A method to reduce the capacity of this uncertainty region is to incorporate more measurement information through additional reference beacons. This process is shown in Figure 4.3, where the addition of an extra reference beacon provides more information concerning the position of the rover node, this further results in the reduction of the uncertainty region surrounding the position estimation.

4.2.2.1 Angle of Arrival

The angle of arrival (AOA) technique for localisation using wireless communication devices such as UWB includes the calculation of the bearing between reference beacons or reference nodes [52]. When providing bearings concerning any aspect of geometry, there is a necessity to provide an object or location from where this angle is referenced. In the case of the AOA technique, AOA provides the capability to acquire the bearing from a node to all other neighbouring nodes in relation to the reference axis of the node that the wireless signal originates from [52, 53]. This reference axis of the node is also known as the reference

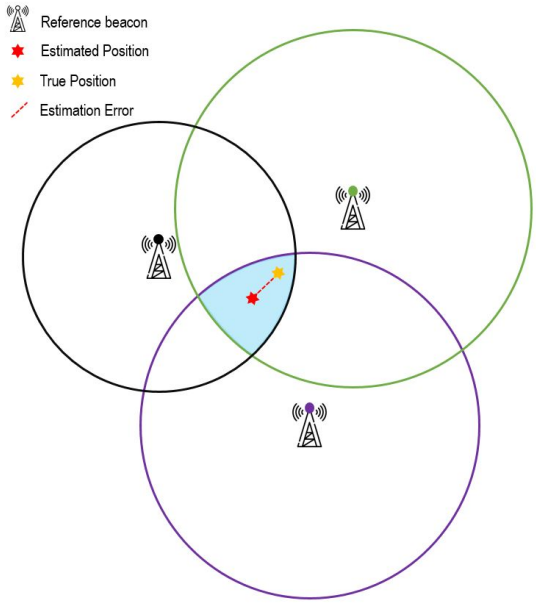


Figure 4.2: UWB positioning using three Beacons

Figure 4.3: UWB positioning using four beacons

direction and is further defined as orientation. Orientation in terms of AOA theory is defined as a fixed direction against which the AOAs are measured, this metric is measured in degrees in a clockwise direction from North [52, 54]. AOA capability is usually achieved using complex antenna arrays and the systems performance is influenced by the geometry of such antenna arrays [53]. The AOA process can be seen in Figure 4.4, where each node has its internal reference direction or orientation from which the AOA is measured [55].

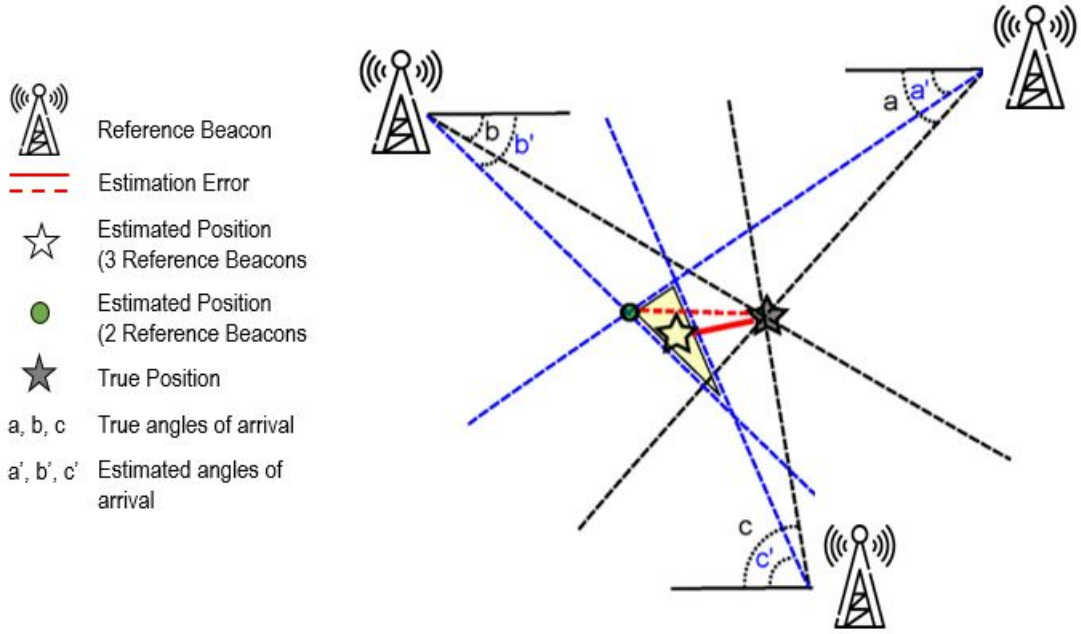


Figure 4.4: Angle of Arrival Technique [55]

It is to be noted that the study presented by *Gezici et al.* state that the use of the AOA approach for localisation for UWB networks is not a suitable implementation. This is due to the increase in antenna arrays in respect to size and quantity and this effectively annuls a main advantage of UWB systems being a lightweight and cost-effective solution [49]. Additionally, *Gezici et al.* states that due to the large bandwidth that UWB operates within, the number of signal paths from node to node within scenarios where the operational environment is cluttered and indoors will affect the angle estimation. This is due to possible scattered objects that are situated within the area enclosed by the reference beacon nodes, possibly reflecting UWB signals and providing bias angle estimations [49].

4.2.2.2 Time of Arrival

The time of arrival (TOA) technique calculates an estimated range using the one-way propagation time between two nodes [55]. Range measurements produced from UWB are achieved from the accurate resolution of the TOA, also referred to as TOF, of a pulsed waveform that is travelling between a set of reference beacon and rover node [50]. Furthermore, as a result of the large operating bandwidth, the UWB system obtains a high time resolution; therefore, positioning techniques can provide accurate range estimations [50]. As demonstrated in Figure 4.5, range estimation from each reference beacon can carry a certain degree of error due to various occurrences such as system perception

limitations and environmental conditions. Beacon deployment in configurations such as that in Figure 4.5, where there are three beacons present is known as the trilateration approach. Figure 4.5 displays markers which denote the true and estimated position, where the estimated position falls into a small shaded area that is created by the distances provided from each beacon. This demonstrates that the position estimation provided by the UWB positioning system can be anywhere within this area. This is due to each range measurement containing a small amount of error. However, this shaded region can be reduced, further reducing the uncertainty within the position estimation by deploying an increased number of reference beacons. This can be seen in Figure 4.6, where four beacons are deployed. This configuration is known as multi-lateration.

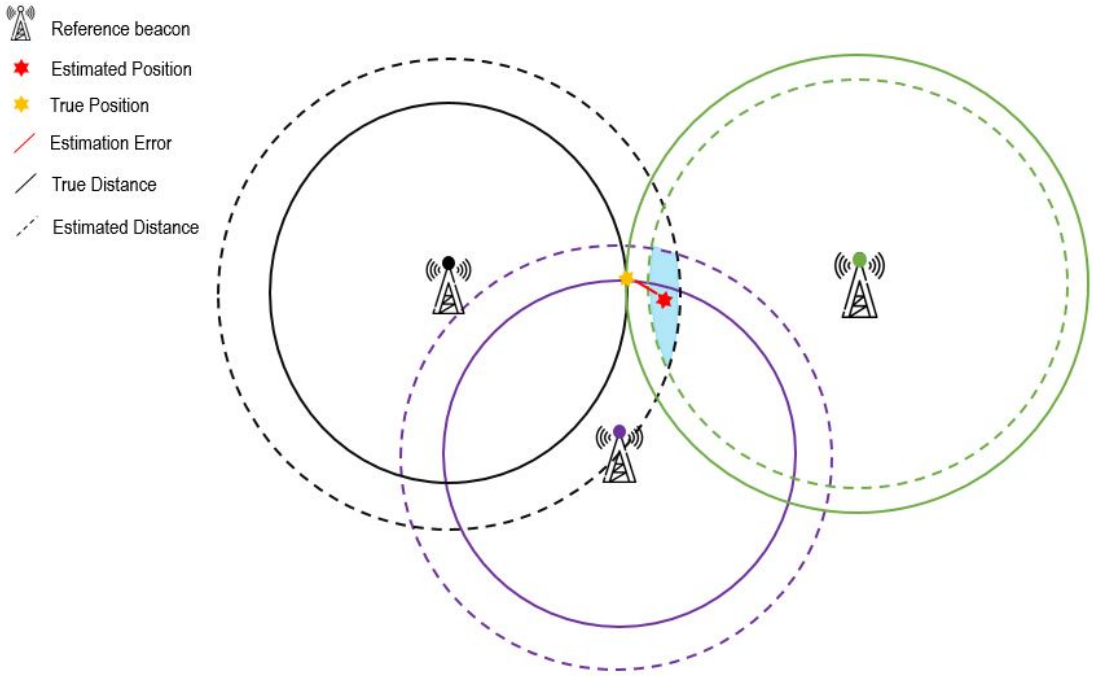


Figure 4.5: UWB positioning using three beacons with estimated and true position

Static characterisation of range uncertainty behaviour between two static UWB nodes has been previously investigated [37]. In order to calculate the range uncertainty for the work carried out in [37], static range readings were taken at various distances between two nodes and compared to a true distance for error calculations, where the true distance was acquired using a laser range finder. Although this method allows static based range error characterisation, it does not provide a dynamic error reading of the UWB system under the intended operating conditions of the deployed system. However, to obtain a more accurate

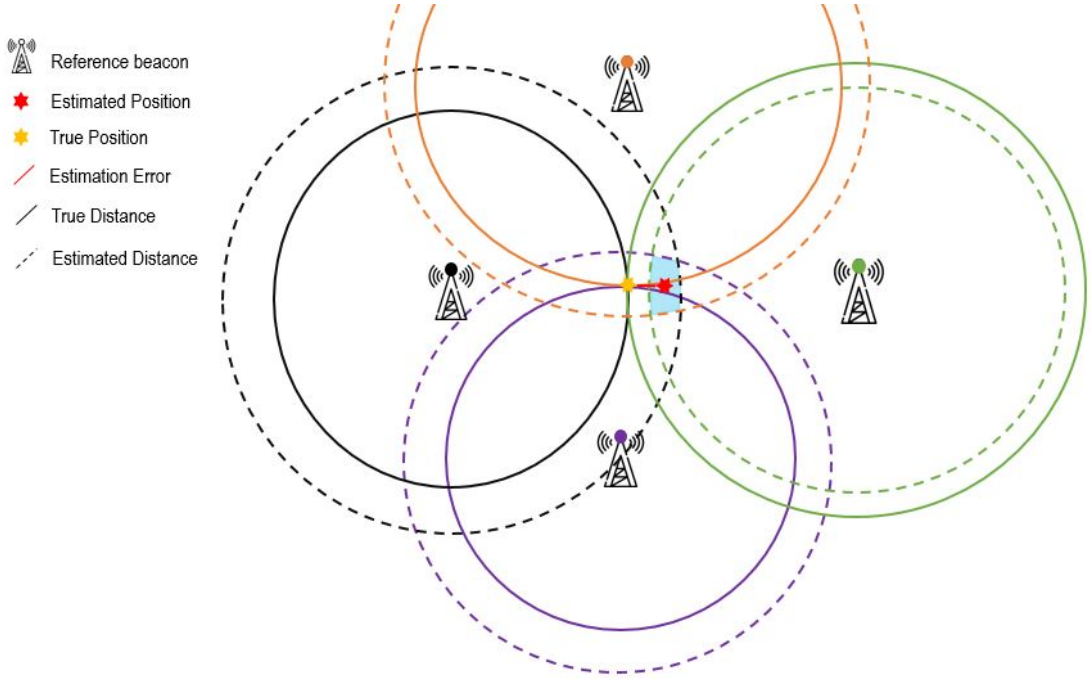


Figure 4.6: UWB positioning using four beacons with estimated and true position

judgement of the expected range error, the system would need to be monitored over various types of movement in continuous rather than discrete samples.

Deploying robotic systems with a means to make judgements and interact with the surrounding environment is entirely dependent upon an imperative element of robotics, which is uncertainty [12]. Uncertainty can arise from factors such as unpredictable environments, actuation factors, inaccurate system modelling and measurement noise produced by on-board sensors. However, for the case of Chapter 4, the range uncertainty of a network of sensor nodes from a UWB localisation system is characterised using a surveying grade optical instrument in the form of an RTS

4.3 Dynamic Uncertainty Characterisation Technique

Precisely quantifying the errors of a sensor or a network of synchronised sensors can be difficult without the availability of a reference figure, often referred to as a ground truth metric. It is therefore vital that the accuracy of the ground truth is of a higher order of magnitude than the system that is characterised. In the design of an EKF, the covariance measurement matrix is required to be known. This measurement matrix represents the

uncertainty of the sensors providing observations into the system, therefore, using a method which can obtain a high degree of accuracy is paramount to acquiring the uncertainty of the sensory units. Therefore, the following subsections describe the methods implemented to characterise the uncertainty of the sensory systems deployed in this stage of the thesis.

4.3.1 Translation Motion Estimation

The components of the 2D pose estimations concerning translation motion (x, y) were studied using a surveying grade optical instrument in the form of an RTS. The RTS was operated with a feature known as Active Tracking, where the RTS would remotely provide updates concerning the 3D translational motion of the mobile robot via an interaction with an optical or reflective target which is mounted on-board. In this case, however, only the 2D output was utilised. For the error characterisation processes, the RTS is used for two main tasks. Firstly, to characterise the uncertainty of sensor systems tasked with providing updates for translational motion of the robot. This uncertainty is then used as input into the EKF as noise covariance matrices Q and R. Secondly, the RTS is also used to quantify the performance output for translational components of the EKF. The remainder of Section 4.3.1 introduces the RTS in terms of the main components, operation features and functionalities, how these surveying grade instruments are deployed in their traditional environments and their integration with a mobile robot for real-time active motion tracking.

4.3.1.1 Robotic Total Station (RTS)

A total station is an optical electronic instrument used to remotely acquire horizontal and vertical distance and angle measurements to a remote point target, where horizontal and vertical angles can be referred to as azimuth and elevation respectively. In its traditional operation, a total station is deployed as a surveying tool in the design of building structures to globally reference the position of building vertices, monitor the structure's changes over time and topographically map the terrain prior to construction. However, to quantify the uncertainty of the unmanned agent's positioning systems an RTS was deployed. The underlying difference between a standard total station and the robotic variant is the ability for the robotic system to be remotely operated through a radio-linked controller device, whereas a total station is operated manually. Also, the RTS possesses active tracking functionalities where it can be configured to continuously and autonomously track moving

targets, a feature which is heavily utilised in this thesis for uncertainty characterisation of the position of the unmanned system. Figure 4.7 shows an image of the Trimble S7 RTS and the TSC3 remote radio-linked control device.

The two main and vital components of an RTS are the electromagnetic distance measurement (EDM) source and the electronic theodolite. These components provide the RTS with the ability to acquire precise distance and angular measurements to targets and points of interest. Such components, along with the mathematical formulation of the method undertaken to convert the acquired distance and angular measurements into a Cartesian format, are discussed below.

Electromagnetic Distance Measurement

The EDM source within an RTS is the component that determines the distance to a remote target in relation to the position of the RTS. The EDM source is a pulsed laser diode which emits a beam of infrared light that is modulated at a controlled rate. The emitted light source is reflected off an external object such as an optical prism and returned to where it originated. Once emitted, the time of flight (ToF) of the pulsed laser is used to

(a) (b)

Figure 4.7: RTS and TSC3 radio linked controller (a) RTS (b) RTS & TSC Controller 3

calculate a distance, this is outlined below.

Time of Flight

In the direct ToF technique, the emitted and reflected beams are both measured and timestamped by a rapidly operating internal clock. When emitted, a beam splitter directs some fraction of the light to a photodetector such as an avalanche photodiode, whose output pulse triggers a begin sequence to the clock. As the laser is reflected, it returns to the receiving component which is also a photodiode where the output pulse triggers a stop sequence, signalling the end of the flight of the laser in that pulse cycle. The overall time t is then related to the finite speed of light in the air medium v, to calculate distance d. The constant v is calculated using the metric for the speed of light in a vacuum c and the air medium refractive index n. This process is identical to that of the LIDAR, where the only underlying contrasting feature is the data acquisition rate. The distance measurement calculation through the ToF technique is found in Equation (4.1).

$$d = \frac{v\ t}{2} \tag{4.1}$$

Where;

$$v = \frac{c}{n} \tag{4.2}$$

- $c = 299, 792, 458$ (m/s)
- $N = 1.0003$

Electronic Theodolite

Theodolites are precision instruments that are implemented to measure horizontal and vertical angles between two points of interest. A theodolite is comprised of a moveable telescope mounted within two perpendicular axes. The telescope operates on both a horizontal and vertical circle to provide angular measurements on both axes. The horizontal and vertical angle measurement scenarios are explained below, accompanied by Figure 4.8 which shows the general structure of a theodolite and its operation on the horizontal and vertical circles.

- Horizontal − Measuring angles on the horizontal circle, otherwise known as the azimuth, between Point A and Point B is achieved through using the telescope to

initially focus on Point A, then applying an azimuth zero offset from that point. Once the azimuth is offset at zero, the theodolite is then orientated to Point B where the angle is changed incrementally until the telescope is directly focusing on Point B. The change in azimuth displayed on the digital screen is the difference in horizontal angle between Point A and B. This is carried out in a local coordinate frame with Point A as a reference.

- Vertical − When the telescope is directly perpendicular to the horizontal plane as shown in Figure 4.8 (b), this is the 0° point on the vertical circle. Measuring angles to a point on the vertical plane, otherwise known as the elevation angle, is achieved through focusing the theodolite directly to Point A. The output angle on the digital screen is the elevation angle of Point A in relation to the 0 point on the vertical circle. The same process is carried out on Point B to acquire the difference in elevation angle between the two points.

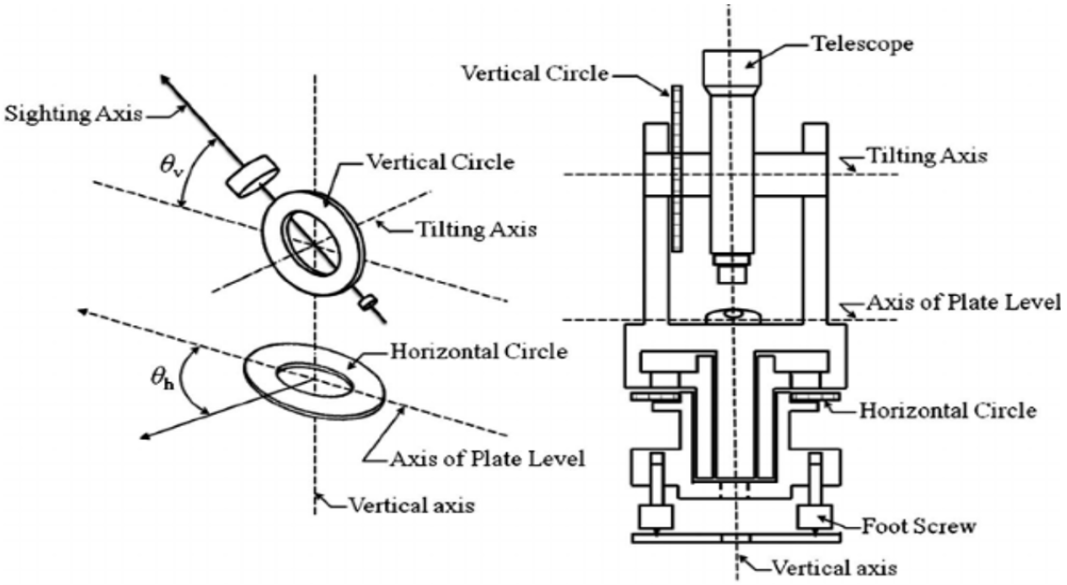


Figure 4.8: Theodolite structure [56]

Polar to Cartesian Conversion

When an RTS provides measurements concerning a static or dynamically situated target, the output is comprised of distance and angle measurements only. Therefore, in order to acquire the Cartesian co-ordinate of the target in relation to the RTS, such distance and angle measurements are required to undergo a conversion which calculates the 3D

Cartesian position. This method is known as the Polar to Cartesian conversion and consists a trigonometric calculation which converts distance and both horizontal and vertical angles to x, y and z. This process is graphically shown in 3D in Figure 4.9, where the equations implemented to calculate each component of the 3D Cartesian co-ordinate are shown in Equations (4.3) to (4.6).

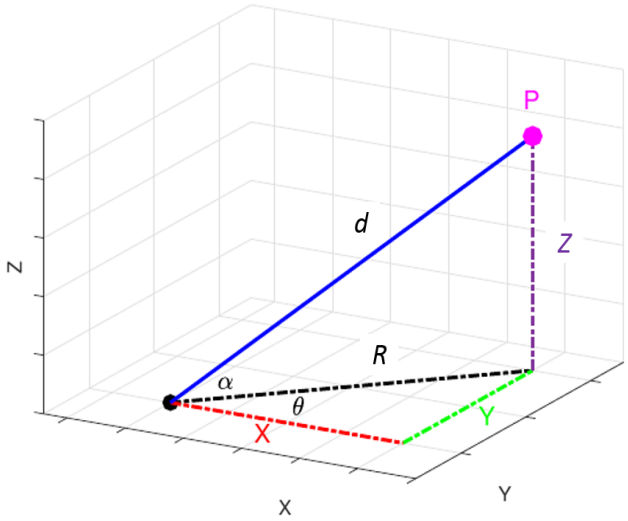


Figure 4.9: Polar to Cartesian Transformation

$$x = d\ \ cos(\alpha)\ \ sin(\theta) \tag{4.3}$$

$$y = d\ \ cos(\alpha)\ \ cos(\theta) \tag{4.4}$$

$$z = d\ \ sin(\alpha) \tag{4.5}$$

$$R = d\ \ cos(\alpha) \tag{4.6}$$

Where;

- P – Measured Point
- d – Distance to P (m)
- θ – Azimuth (horizontal angle) to P (rad)
- α – Elevation (vertical angle) to P (rad)

- x − Calculated Cartesian x co-ordinate of P
- y − Calculated Cartesian y co-ordinate of P
- z − Calculated Cartesian z co-ordinate of P
- R − 2D Distance to P (m)

4.3.1.2 Trimble S7 Robotic Total Station Specifications

The RTS takes measurement readings using an optical and remote distance sensing unit with an expected accuracy of 2mm + 2ppm in standard prism mode and 4mm + 2ppm in tracking prism mode. The difference between standard and tracking prism modes is that the standard method is a direct shot to a static target, and the tracking alternative is when the target is dynamic and the RTS is set to actively track the target. In this section, expected performance parameters are provided for the Trimble RTS. The performance characteristics are categorised into two sections, the EDM source for distance measurement and the absolute encoder, which enables the theodolite to provide angular measurements.

Details acquired from the manufactures specification document concerning component types and operational performance of the RTS can be found in Table 4.1.

Table 4.1: RTS Specifications

Performance		
Distance Measurement	**Accuracy (ISO)**	±1mm + 2ppm (Standard Mode)
	Root Mean Squared Error (RMSE)	4mm + 2ppm (Tracking)
	Range	0.2 - 2500m
	Update Time	1.2s (Standard Mode)
	Update Time	0.3s (Tracking Mode)
Angle Measurements	**Sensor Type**	Absolute encoder
	Accuracy (Std. Dev.)	1" or 2.77×10^{-6} degs
Component Specifications		
EDM Source	**Light Source**	Pulsed laser diode
	Wavelength	905nm
	Laser Class	Class 1 (eye safe)
Communications	**Interface technique**	Bluetooth Module RS232 Serial
		Radio Control Link

4.3.1.3 Active Target Tracking

When deploying the RTS as a method for characterising the uncertainty behaviour of different positioning systems, the RTS requires a reference frame to operate within. Simply outputting and utilising the raw distance and angle measurements will provide useful information, however, there needs to be a pre-defined reference co-ordinate frame to enable a metric system for error characterisation. In this case, the distance and angle measurements from the RTS were converted into a Cartesian co-ordinate system with the position of the RTS as the origin. The sections below outline the RTS configuration procedures and its interaction with mobile unmanned robots.

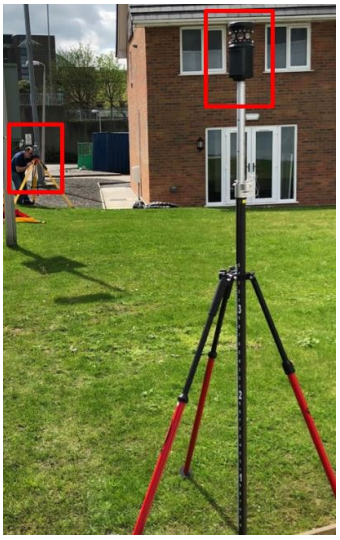
RTS Configuration

For all testing procedures using the RTS throughout this thesis, the following calibration

procedure was consistent and recursive to provide validity for all experiments.

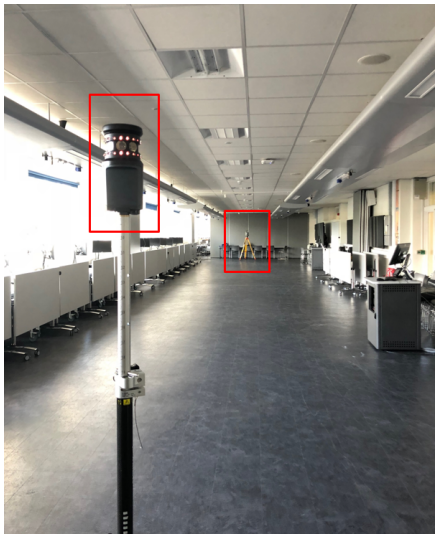
When tracking an unmanned system, there needed to be a definition of a global operating frame. In this case, the global frame was taken to be the local frame of the RTS. This therefore means that all acquired measurements from the RTS are referenced from the local position of the RTS and the RTS is not set to have a geo-referenced position within a co-ordinate system such as Word Geodectic System 84 (WGS-84). For all experiments, the RTS was set to have a position of $RTS_P = [100,\ 100,\ 10]^T$ to prevent readings crossing into the negative sections of each axis. In order to define the axis components of the global frame, a calibration procedure was undertaken using a back-sight observation. The back-sight observation is a measurement taken to an optical 360° prism which is attached to a centrally mounted telescopic pole and this measurement sets the angle zero position of the absolute encoder on both the horizontal and vertical planes, therefore setting the Y and Z axes. The back-sight measurement procedure can be seen in Figure 4.10 and the outputted results from the back-sight calibration is shown on the radio-linked TSC3 controller in Figure 4.11.

(a)



(b)

(c)



(d)

Figure 4.10: Back-sight measurement for azimuth reference (a) 360° IR-LED optical prism (back-sight) (b) Outdoor testing area (c) Indoor testing area RTS point of view (POV) (d) Indoor testing area back-sight POV

Unmanned System Tracking

For the purpose of actively tracking the unmanned agent, there was a necessity to provide a linkage between the mobile robot and the RTS. This was achieved through mounting lightweight and robust reflective and optical targets to the robot which ensured that the RTS would consistently be able to track the robot's position, this process is also known as an optical lock, where the electromagnetic distance measurement source provides a continuous tracking feature. Therefore, this further initiates autonomous absolute encoders control to allow the head of the RTS to follow the movement of the robot. Initial methods that were implemented utilised a circular 180° prism which was mounted to the rear end of Version

(a) (b)

(c)

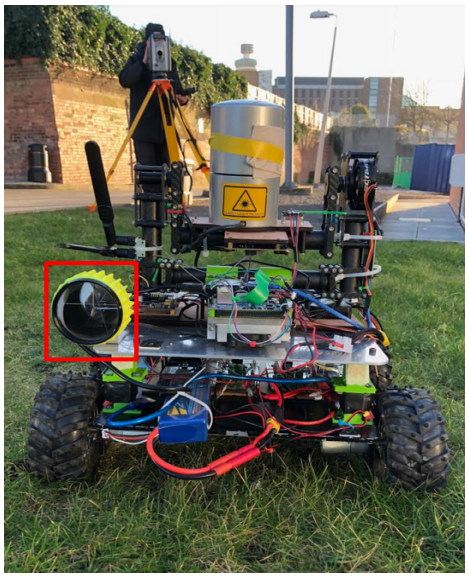
Figure 4.11: Back-sight Observation Results (a) Horizontal and vertical zero reference offsets (b) Horizontal and vertical distance, (c) 3D position of the RTS

1 of the developmental platform shown in Figure 4.12a. However, this initial method experienced limitations as the trajectories that the robot could execute were limited due to the RTS losing of sight of the circular 180° prism when the robot executed a sharp turn or attempted to execute a circular motion. Test trajectories were therefore limited to a

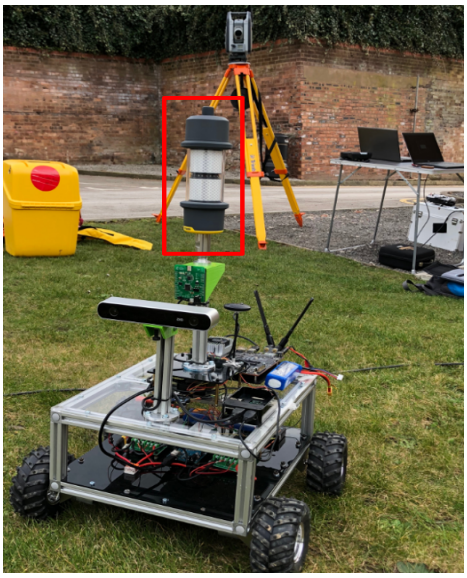
linear nature or very simplistic curves. However, this limitation was overcome through the introduction of a 360° reflective target which was centrally mounted in an advanced position and therefore provided a consistent clear line of sight from the RTS to the robot. The 360° reflective target is a battery powered device which deploys high frequency light emitting diodes (LEDs) to enhance the active tracking capabilities of the RTS. This reflective target can be seen in Figure 4.12b on-board Version 2 of the robot. Although the 360° reflective target overcomes the robot trajectory limitations held by the 180° prism, the 360° target is inefficient in terms of payload size and weight. This had a resulting effect on the robustness of deployment of the robot and the majority of the maximum payload weight was assigned to the 360° reflective target. Therefore, Figure 4.12c shows the implementation of a miniature 360° optical prism which enables both sophisticated trajectory tracking and flexibility in terms of payload weight and placement. Figure 4.12b as a whole demonstrates the progressions made over time in relation to platform design and a balance between ease of deployment and payload considerations.

4.3.1.4 Robot Trajectory Plotting using an RTS

To demonstrate the effectiveness of using an RTS as a metric for a ground truth, trajectories of the robot are plotted below using the three presented target solutions. The trajectories are represented by a solid line which shows the nature of the robot movement and hollow points denoting a measurement update from the RTS which provides updates at 3Hz. The update frequency cannot be altered as it is a manufactured characteristic. The tracked trajectories shown in Figures 4.13 to 4.15 show the noise-free behaviour of the position updates and the smoothness of the tracked trajectory of the robot. However, Figure 4.13 shows the limitations of the 180° reflective target circular prism in terms of trajectory simplicity in comparison to the trajectories shown in Figure 4.14 and Figure 4.15 from the 360° solutions. In order to fully test the precision of the positioning systems designed to operate on-board the robot, multiple scenarios both simplistic and of a higher level of difficulty in terms of a planned trajectory must be executed to fully understand the nature of the error and performance in contrasting situations. Therefore, throughout the rest of this thesis, the 360° targets were used to characterise the uncertainty of the positioning sensors in question, in specific the miniature 360° optical prism. However, the large 360° reflective target was used for the study conducted in [32].



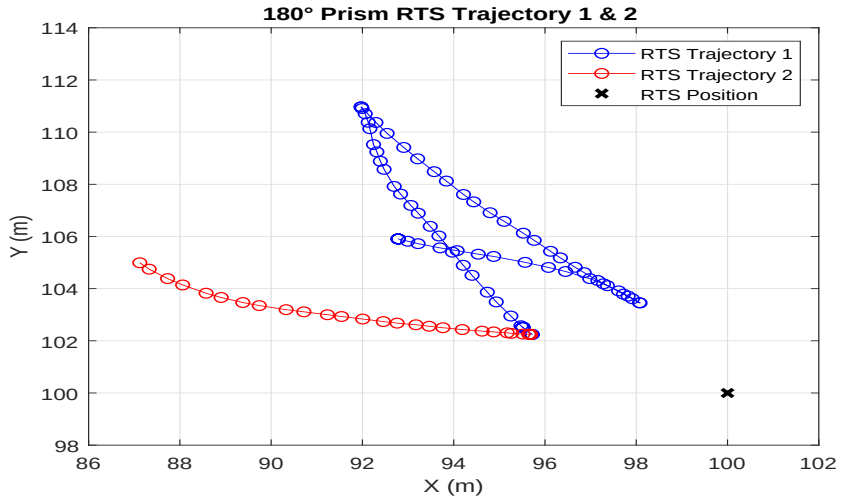
(a)



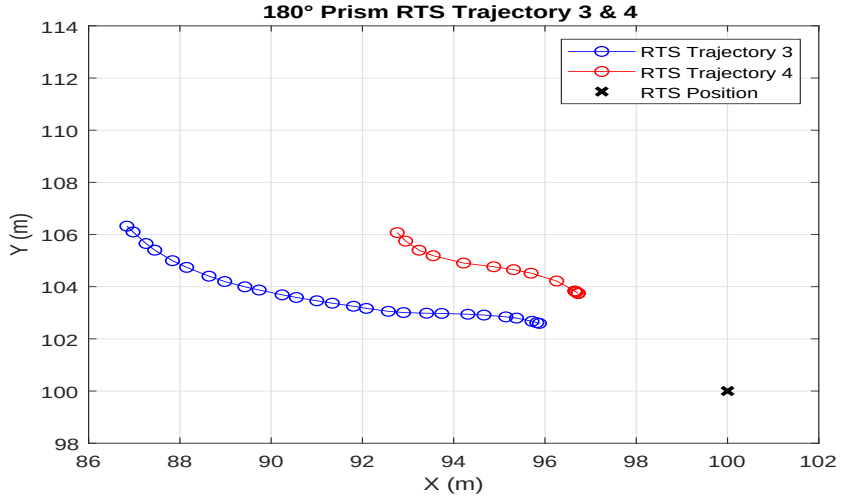
(b)

(c)

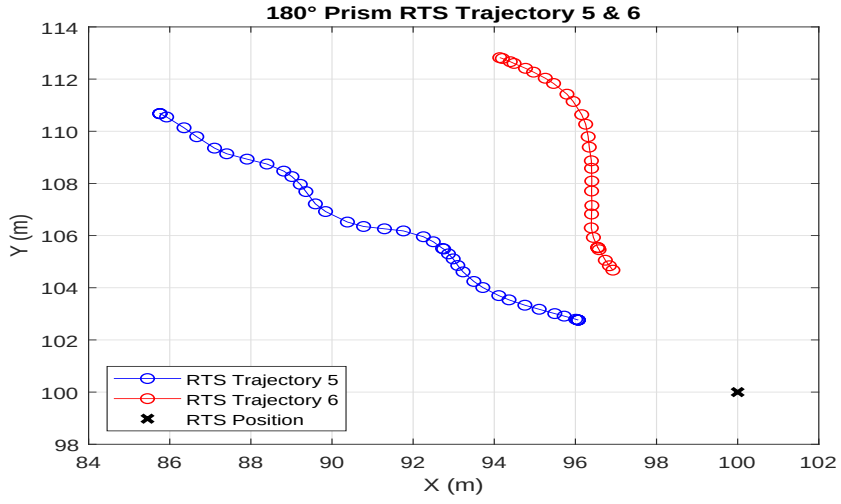
Figure 4.12: Platforms with optical targets mounted (a) Platform 1 with 180° optical prism, (b) Platform 2 with 360° IR-LED reflective target, (c) Platform 3 with miniature and lightweight 360° optical prism



(a)

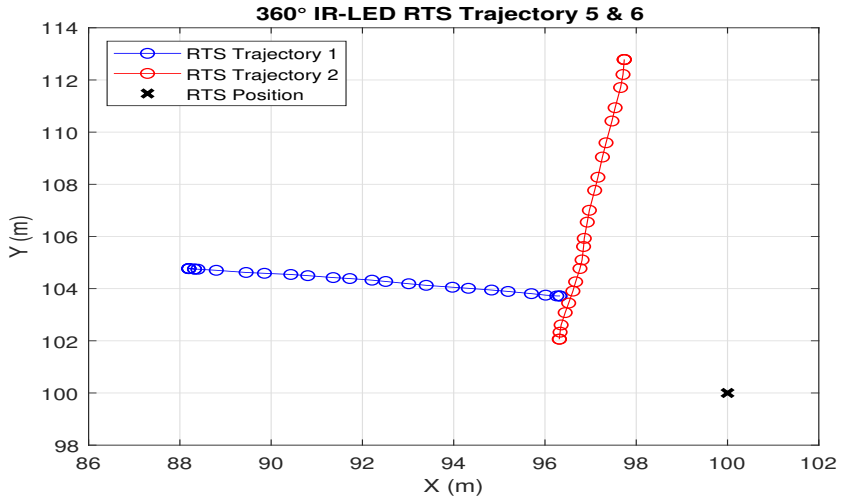


(b)

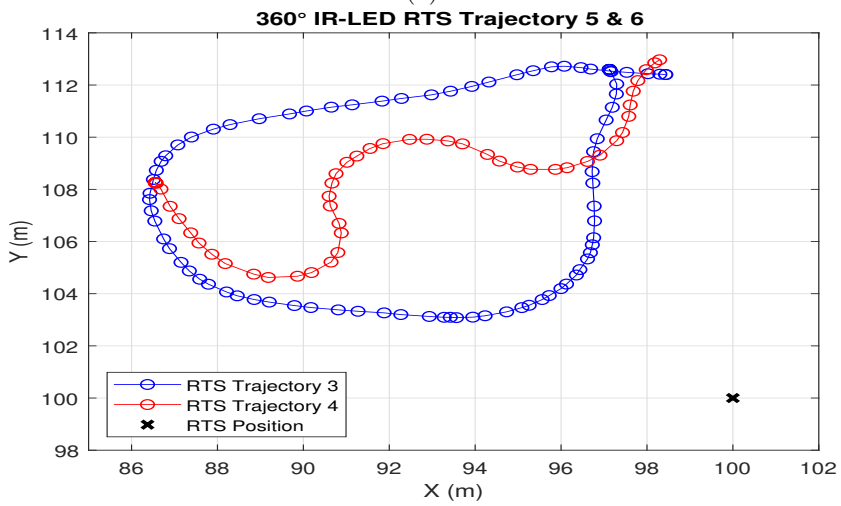


(c)

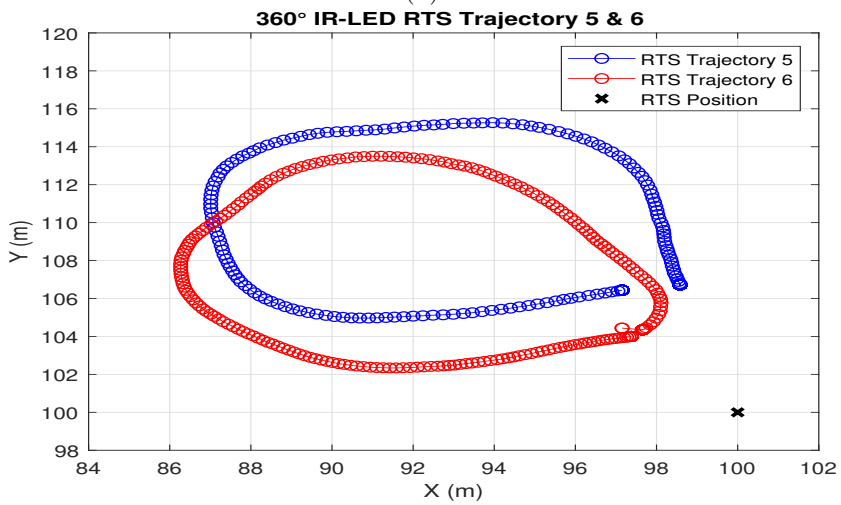
Figure 4.13: 180° Prism RTS Trajectories (a) Trajectory 1 & 2 (b) Trajectory 3 & 4 (c) Trajectory 5 & 6



(a)

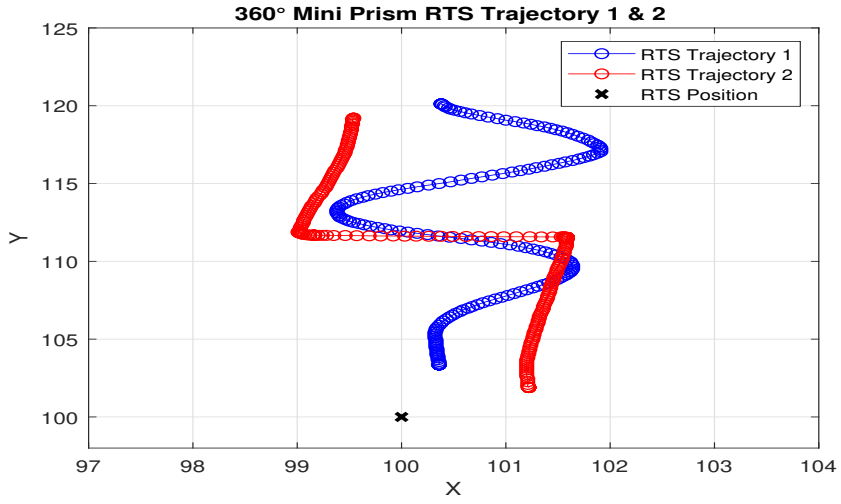


(b)

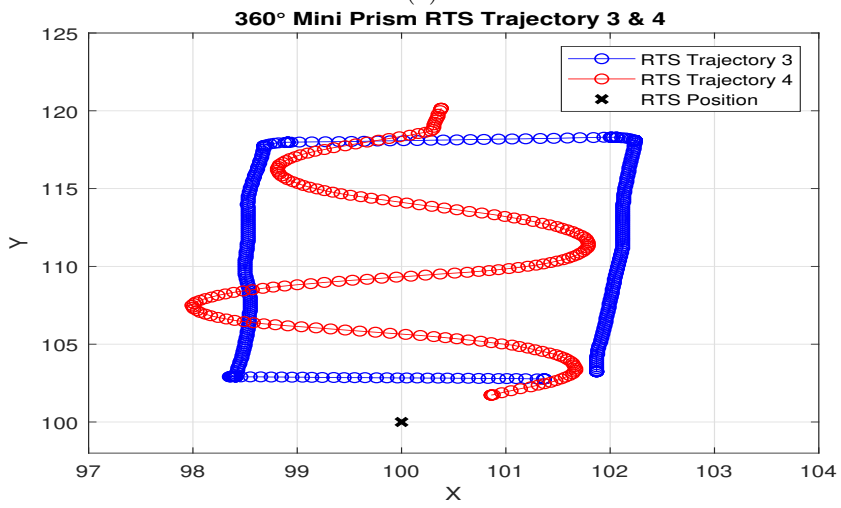


(c)

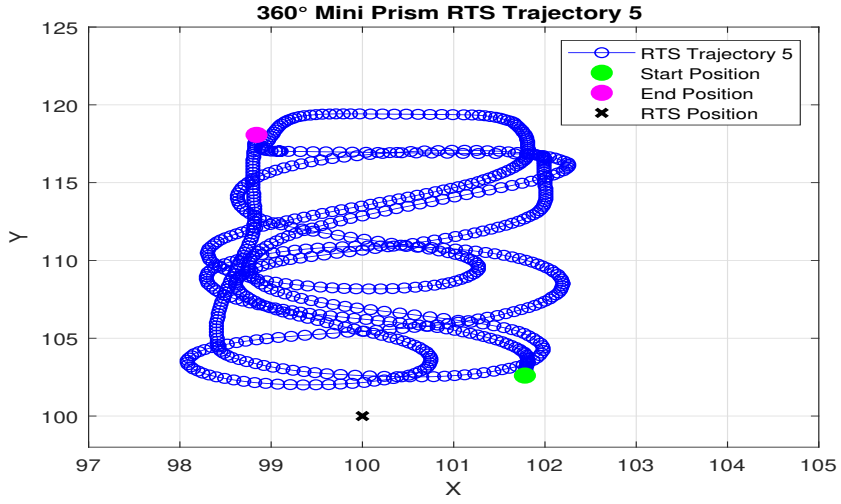
Figure 4.14: 360° IR-LED Target RTS Trajectories (a) Trajectory 1 & 2 (b) Trajectory 3 & 4 (c) Trajectory 5 & 6



(a)



(b)



(c)

Figure 4.15: 360° Miniature Prism RTS Trajectories (a) Trajectory 1 & 2 (b) Trajectory 3 & 4 (c) Trajectory 5

4.3.1.5 Conclusion

This section has presented the methodology undertaken to actively track an unmanned system using surveying grade optical instruments. The following summarises the main conclusions arising from this section of work:

- The RTS provides a smooth and noiseless trajectory with millimetre accuracy. These position outputs acquired from the RTS are deemed as a suitable metric to use as the ground truth for sensor characterisation.
- Although a 180° optical prism is more efficient in terms of size and weight, many limitations arise from its implementation as tracking the robot fails once the RTS loses direct line of sight. This occurs in a situation where the robot executes a sharp turn or enters a circular motion.
- The 360° reflective target with high frequency LEDs provides a reliable solution to allow the RTS to track the robot. The 360° target performs in all types of motion. However, its disadvantage arises from its size and weight.
- The 360° miniature prism provides a reliable solution to allow the RTS to track the robot. It also compensates for both tracking limitations and size and weight issues held by the 180° prism and the 360° reflective target respectively.

4.4 Sensor Uncertainty Characterisation

Section 4.4 presents the methodologies undertaken in order to characterise the error variance of the sensors used in the design of the R-EKF. It begins with the expansion of an IMU characterisation procedure, quantifying the error measurements of a heading rate gyroscopic output. Secondly, Section 4.4 presents the characterisation of the range error within the UWB network presented in Section 4.4.2, in both static and dynamic conditions. The latter experimental condition is where the optical tracking feature of the RTS previously presented in Section 4.3 is utilised.

4.4.1 Gyroscope

Section 4.4.1 incorporates two subsections Section 4.4.1.1 and Section 4.4.1.2 which respectively present the methodology undertaken and the results when quantifying the

uncertainty of the gyroscope rotational rate sensor.

4.4.1.1 Methodological Approach

Due to the rapid update rate of the IMU (50Hz), the ability to dynamically characterise the error variance of the gyroscope as it operates is difficult due to measurement time-stepping and the lack of a sufficient time-synchronised technique to achieve this. Therefore, a decision was taken to characterise the gyroscopic error statically at room temperature. This provided a comparison metric at 0 as the unmanned system was stationary and any output produced from the gyroscope was considered as erroneous measurements and therefore quantifiable. The IMU on-board the unmanned system was considered as a proprioceptive unit, which means that it is a device which provides observations concerning the relative position of the body of the system on which it is mounted. For systems such as an IMU, which encapsulates gyroscopes, accelerometers and often magnetometers are accompanied with manufacturer's bias that needs to be accounted for when processing the sensory output. Therefore, the aim of this sub-experimental procedure was to produce bias estimation for the yaw rate of the heading gyroscope as well as the error variance produced over the course of ten static testing procedures, each procedure aimed at acquiring 10,000 measurements which equates to a 200s testing time period, this is graphically represented in Figure 4.16. All individual static test iterations were compiled to generate a fully concatenated dataset which was then used to quantify the gyroscopic bias and error variance. This is graphically represented in Figure 4.17. Furthermore, the details concerning these test iteration procedures are shown in Table 4.2.

4.4.1.2 Results

Section 4.4.1.2 presents the results acquired from the conducted experimental procedure presented in Section 4.4.1.1. Over the course of the experimental procedure, the gyroscope was assessed in static conditions on-board the mobile unmanned platform. The results shown in Figure 4.18 present a histogram plot concerning the final concatenated dataset. The nature of the histogram demonstrates that the gyroscopic error acquired over the course of the experiment forms a Gaussian distribution. This can be identified through the bell shaped distribution that the histogram bars form.

Additionally, the temporal nature of the gyroscope error can be seen in Figure 4.19.

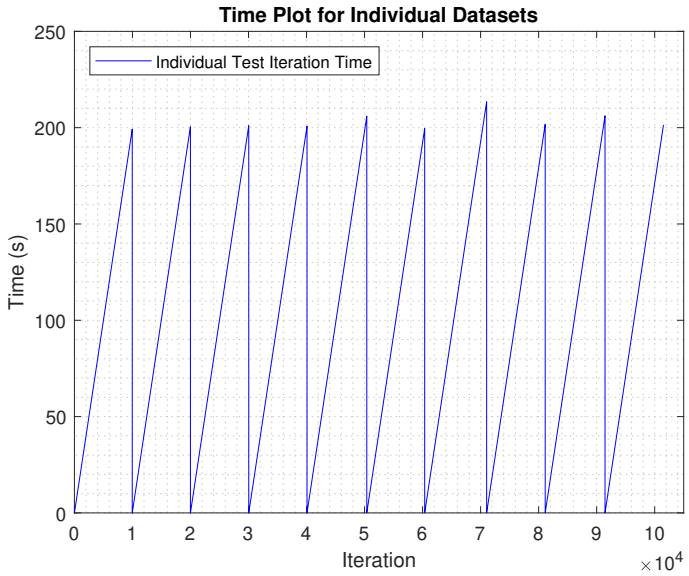


Figure 4.16: Time statistics for individual gyroscope test iterations

Table 4.2: Details concerning static gyroscopic error characterisation procedure

Test No.	**Time (s)**	**No. of Measurements**
1	199.27	9965
2	200.51	10027
3	201.05	10054
4	200.77	10040
5	205.97	10300
6	199.55	9979
7	213.49	10676
8	201.81	10092
9	206.15	10309
10	201.45	10074
Total	**2030.04**	**101516**

It is evident that the gyroscope produced a near 0 mean error distribution over time with a healthy Gaussian like distribution. From this, numerical results were extracted and tabulated in Table 4.3. Certain cells within Table 4.3 includes the acronym *CR*, which denotes the 95% confidence region. The 95% CR states that the PDF acquired from the modelling of the gyroscopic error is 95% certain that the values relating to the mean (μ) and standard deviation (σ) or error fall within these regions. The CR's are presented using

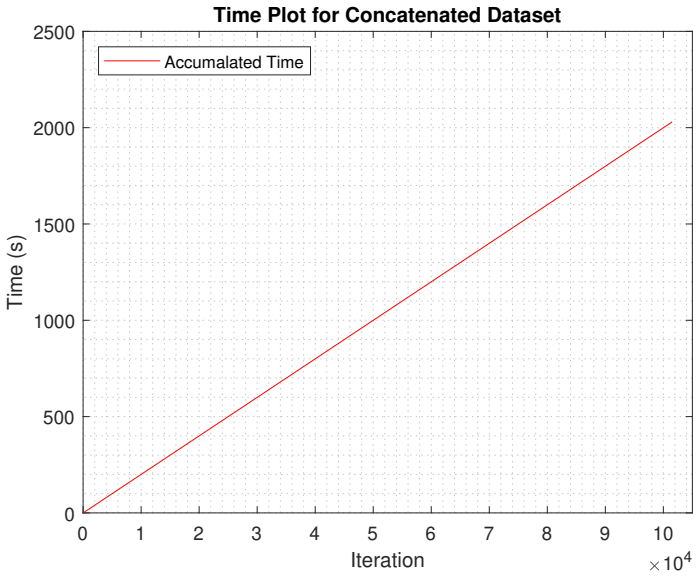


Figure 4.17: Time statistics for concatenated gyroscope test iterations

a hyphen (-) to separate the lower and upper bounds of the CR. This is not to be confused with the mathematical notation of subtract.

Table 4.3: Gyroscopic error characterisation statistics

Mean (μ)	**95% CR** (μ_{CR})	**Std. Dev.** (σ)	**95% CR** (σ_{CR})
-6.64×10^{-7}	-8.04 - $6.71 \quad \times 10^{-3}$	1.199×10^{-3}	1.193 - $1.2043 \quad \times 10^{-3}$

4.4.1.3 Conclusion

In conclusion, Section 4.4.1 presented the method implemented to characterise the bias and measurement error variance of the gyroscope that is utilised in the system as a control input. The gyroscope was seen to experience a near zero mean error of a Gaussian nature with a standard deviation of 1.199mrads/s, therefore enabling the input to be utilised correctly in the design of the R-EKF. It is to be noted here that the large gathered set of gyroscopic data in static conditions was conducted only at room temperature. It is a well-known fact that gyroscopes and accelerometers which are considered as sub-sensors within IMUs, are prone to possess erroneous tendencies when subjected to fluctuations in temperature. This can therefore result in a dynamic alteration of the bias concerning these sub-sensors. In the case of this experiment, temperature assessment was overlooked as the unmanned system was assumed to not experience significant changes in operating temperature. However, it

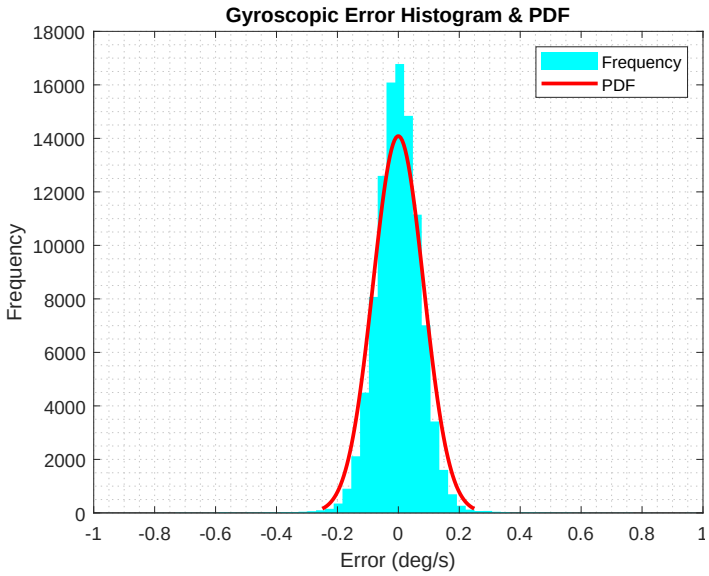


Figure 4.18: Gyroscopic error histogram plot and PDF demonstrating a Gaussian nature

can be considered as a possible future assessment to analyse how temperature fluctuations affect the bias and error variance of the gyroscope.

4.4.2 Ultra Wide-Band (UWB)

Section 4.4.2 presents the range error characterisation for the UWB sensor network. It consists of a static and a dynamic based experimental procedure, which are explained within Section 4.4.2.1 and Section 4.4.2.2 respectively.

4.4.2.1 Static Testing

The initial usage and analysis of the UWB system was conducted through static based experiments. Even though the system was intended to be deployed dynamically on-board the mobile robot, a conclusion was reached to first understand the system in simplistic static conditions. The statically based experiments were conducted within two sections. The first section of testing procedures was conducted in an indoor laboratory corridor environment with and without the inclusion of physical barriers. Lastly, the second section of testing was conducted in a subterranean environment over long ranges to assess how the system behaved in strenuous conditions. These experimental procedures are presented in the following sections.

Static Experimental Procedure 1

The following methodology and results are presented in the International Conference

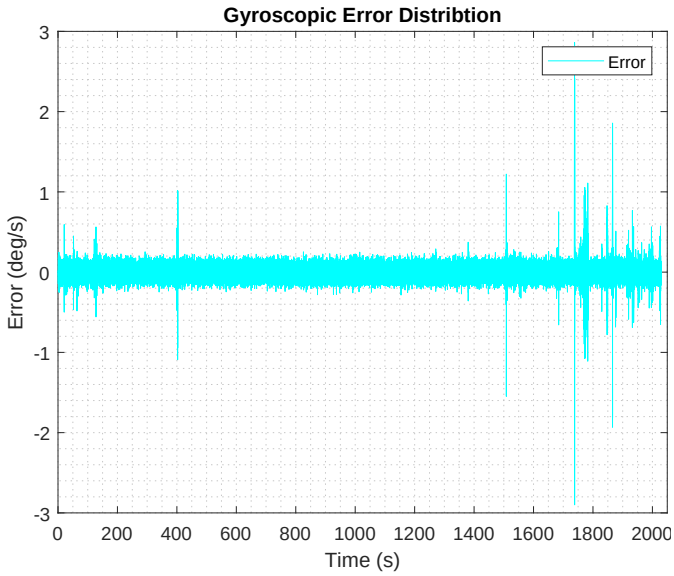


Figure 4.19: Gyroscopic error distribution over time

named *Towards Autonomous Robotic Systems (TAROS), 2018* [31].

Methodological Approach

The first static based experiment consisted of placing two UWB transceivers in a corridor-based environment whilst incrementally increasing the distance and angle between them. This experimental approach was undertaken to identify how the range error between the two devices behaved in relation to an increase in angle and true distance. For this experiment, a hand held miniature Bosch (Portable Laser Ranger) PLR 50 C laser range finder was utilised as the ground truth metric. The PLR 50 C has a ±0.002m typical measurement accuracy.

The nature of movement of the UWB transceivers during the experiment can be seen in Figure 4.20, where the distance between the static and mobile node transceivers was increased to a maximum of 20m at 2m increments. However, post completion of a full 20m track iteration, physical barriers were then included to increase the complexity of the environment and to simulate what is known as non-clear line-of-sight (NCLOS) conditions. This was conducted to assess the behaviour of the UWB signal transmission distortion through physical objects as it is highly likely that when mounted to a mobile system, the line of sight for communication between rover and anchor nodes will be obstructed. The inclusion of these barriers is shown in Figure 4.20.

Each UWB tag transceiver comes equipped with filtering algorithms that fuse data from

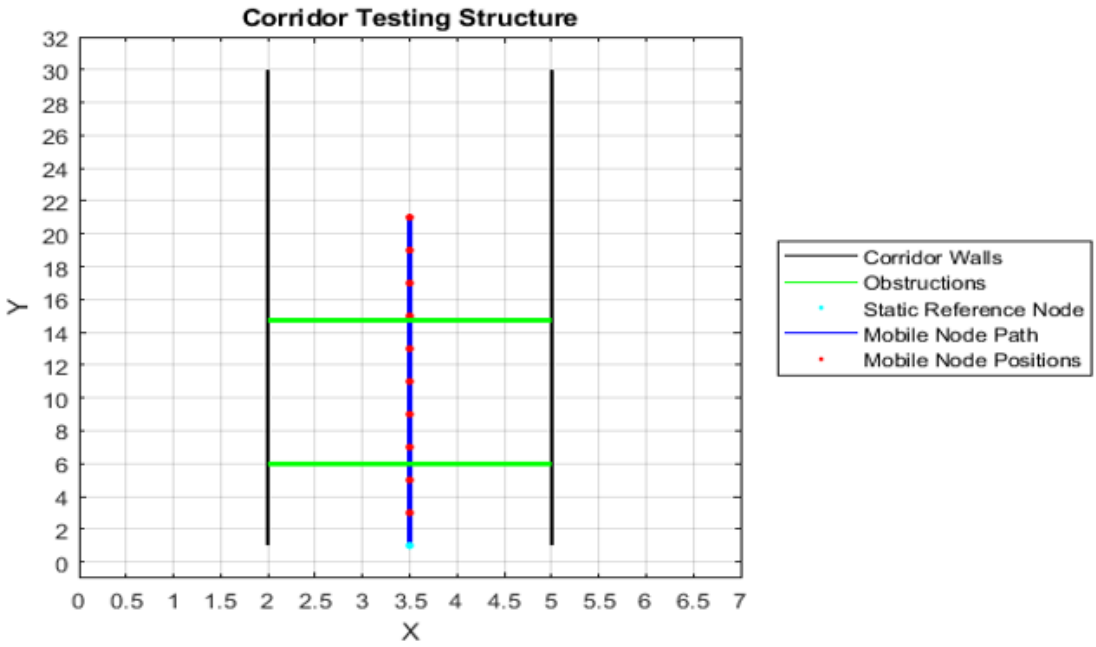


Figure 4.20: Indoor corridor test environment structure

the on-board IMU. This characteristic was used to incrementally increase the pitch angle of the rover node from 0° - 90°. The tested orientations are shown in Figure 4.21.

Therefore, to conclude the description of the methodology carried out in four sets of experimentations, a 20m track at 2m iterations was completed at each of the four orientations. This was done for both clear line-of-sight (CLOS) and NCLOS. The results are presented in the following section.

Results

The results gathered from the experimental methodology completed in the previous section present the range error statistics acquired for the UWB system at various distances and orientations. As the nature of the UWB system consists of a dynamic rover node, in this case intended to be mounted to a mobile robot, communicating with a set of static reference anchor nodes to provide local position estimations of the mobile device, testing the capabilities of the system at various ranges and orientations was necessary. Figure 4.22 shows the acquired error statistics and behaviour of mean and standard deviation of range error in relation to distance and the occurrence of environmental physical barriers.

Firstly, it can be seen in Figure 4.22 that the addition of physical boundaries within the environment in NCLOS scenarios increases range error in comparison to CLOS. This was expected as the transmission of the UWB signal through a medium would result in

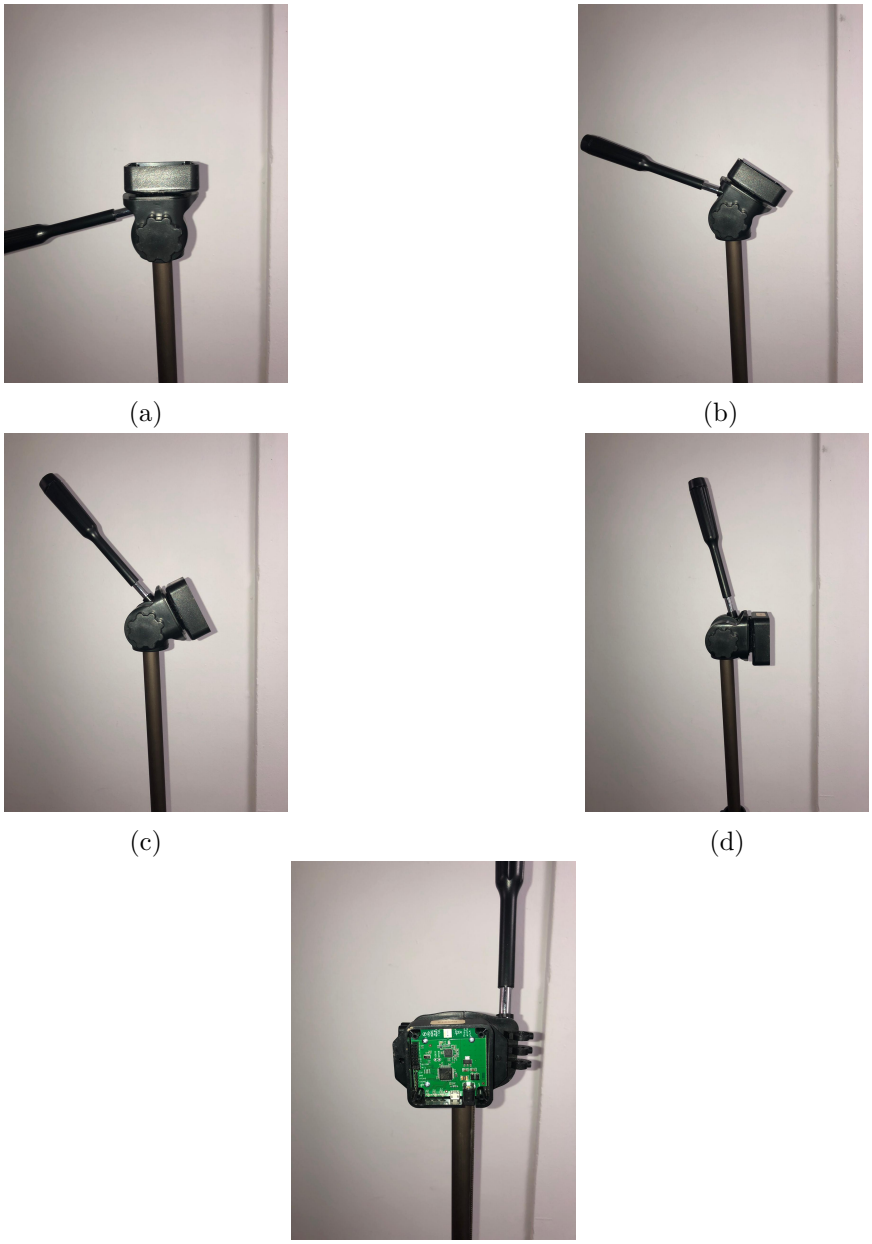


Figure 4.21: Static UWB Range Error Characterisation Device Orientations (a) 0° (b) 30° (c) 60° (d) 90° (e) Front of Static Node

a signal reception delay, therefore distorting the resulting range measurement. Secondly, it is demonstrated that although in NCLOS the mean error at each distance is generally larger than in CLOS, the standard deviation, for both environmental conditions, is similar. It is also notable that for all orientations in both CLOS and NCLOS, the error percentage of true range is seen to remain below 3%, however, the only exception to this finding is at a 2m distance when orientated at 0° and at 4m when orientated at 60°, this can be observed in Figure 4.22a and Figure 4.22c respectively. It was at the orientation of 0° that

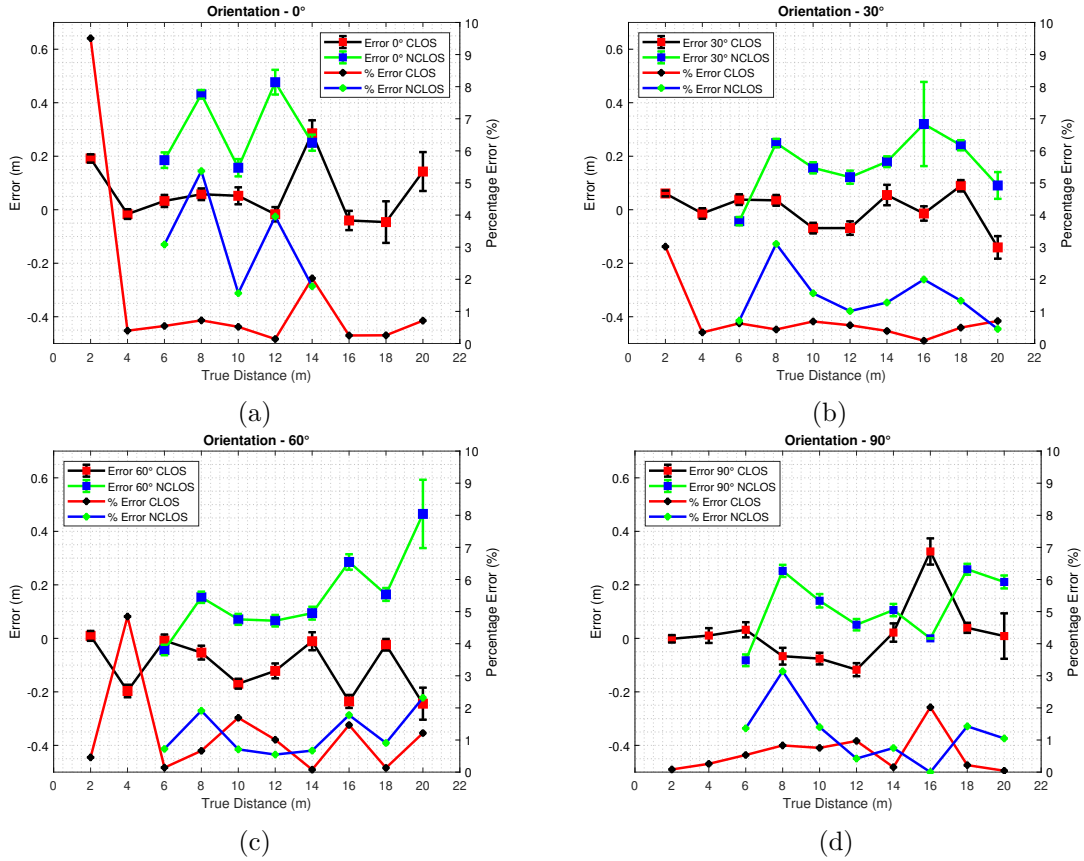


Figure 4.22: Static UWB Range Error Characterisation Statistics (a) 0° (b) 30° (c) 60° (d) 90°

the two devices failed to communicate within NCLOS conditions. This is noticeable at 14m in Figure 4.22a, this demonstrates a possible limitation of the system. When assessing all error statistics within the four orientations, it is clear that when the tag is orientated at 90° to the anchor, the range error for both CLOS and NCLOS is lower and also results in a lower overall percentage error. This can be observed in Figure 4.22d and therefore suggests that when mounted to the mobile system, the tag should be orientated at 90° to provide a reduced overall range estimation error. The final and averaged mean range error statistics in both conditions are shown in Table 4.4 further concluding that at 90°, the two UWB devices produce better range estimations over a span of 20m.

Post completion of the experimental methodology and data processing presented above, certain issues and uncertain areas concerning the results and testing procedure arose. These issues originated from the complete absence of communication between the devices when orientated at 0° and at a NCLOS distance of 14m. Mean error and standard deviation of error were presented at each distance, however, from the resulting datasets it was noticeable

Table 4.4: Mean error statistics for static corridor testing

Position	**CLOS** μ (m)	**NCLOS** μ (m)	**Iterations (at each distance)**
0°	0.083	0.24	500
30°	0.065	0.138	500
60°	0.1	0.13	500
90°	0.048	0.11	500

that the UWB system was experiencing occasional time-outs and loss of communications that were not registered during the testing procedure. This was an issue as it generated misleading impressions regarding the range error behaviour. For example, focusing on the statistics concerning the 30° orientation shown in Table 4.4, the results gathered state that at 18m there is a mean error of approximately 0.05m. Hypothetically, it is possible that 75% of the total 500 measurements acquired at 18m were in fact drop outs in communications, and the statistical error calculations were only conducted on 25% of the overall intended measurement sample size. This resulted in the creation of an inconsistency in experimental procedures. Therefore, an experiment of a similar nature was conducted whilst taking into account signal dropouts and lapses in communications. This is presented in the following section.

Static Experimental Procedure 2

Methodological Approach

The results presented in the first static experimental procedure concluded with the occurrence of an unknown factor that created misleading interpretations of the acquired range error statistics. The negligence of monitoring the signal transmission dropout rate between the UWB devices generated some uncertainties relating to the results, especially as the distance between the devices increased. Therefore, a second static based experimental procedure was undertaken in a similar nature. This procedure was conducted in a large scale environment to increase complexity, where the communications and range estimations of the UWB system were tested between 2m - 44m, this time increasing the sample size to 1,500 measurements and without the inclusion of NCLOS conditions. The first static

experiment proved that the devices experienced optimal performance at 90°, therefore both UWB devices were maintained at 90° throughout this second experiment. It is notable that during this experiment, the percentage dropout of communications was monitored to analyse its relationship with the increase in distance. The results are presented in the following section.

Results

Similar to the results presented in the first static experimental procedure, the findings acquired for this experimental procedure also show the relationship between mean and standard deviation of error at each distance but on a larger scale. It is shown in Figure 4.23 that the initial stages where the true distance is between 2m-6m, the mean error is seen to be constant at just below 0.2m and the standard deviation is minimal. However, fluctuations in range error are seen to occur between 8m-14m before it is then seen to average out at around 0.2m mean error. As previously mentioned, results in Figure 4.23 show the occurrence of communication failures, where the devices register a zero metre range estimation. It is evident that at each testing distance there are occurrences of communication lapses, albeit on a small-scale between the distances of 2m - 30m. However, the percentage of measurements that experienced communication lapse increases after the 30m mark, thereby demonstrating that the communications within the system become strained after 30m. Therefore, the concerns expressed in the final stages of the first static experimental procedure were confirmed during this experimental procedure, as although the system produces a low mean error at larger distances, the percentage of communication dropouts is relatively large. An effective reference for this statement can be seen at 38m and 44m, where the mean error and standard deviation of error is relatively small in relation to the true distance, however, nearly 90% of the total acquired readings were in fact lost due to communication lapses.

The results presented in Figure 4.24 show a 3D plot demonstrating the relationship between the true range, standard deviation of error and percentage of signal dropouts. When designing state estimation sensor fusion algorithms, the standard deviation of the system error is required in the design of the noise covariance matrices. The standard deviation of error is seen to be generally low between 2m - 30m, although there are outliers present between 12m - 20m. However, as the percentage of communication dropouts

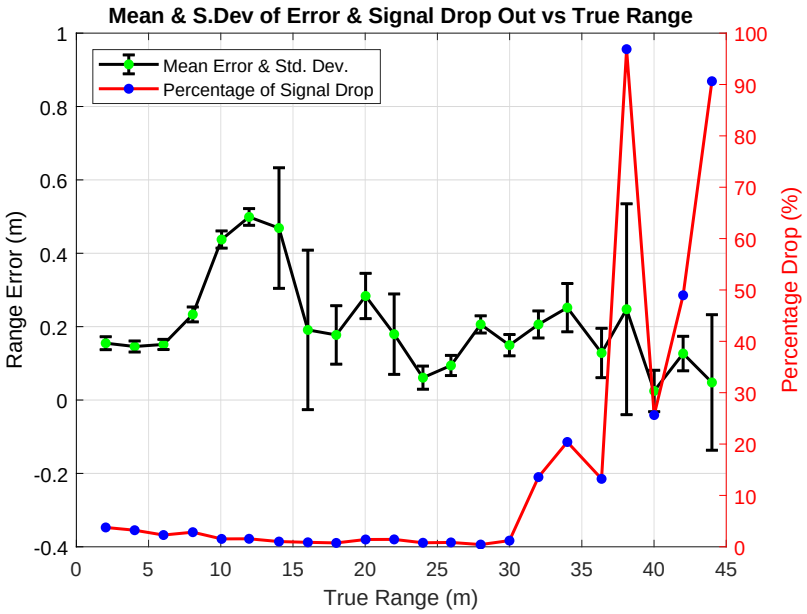


Figure 4.23: Error statistics for corridor long range tests

increases, the standard deviation is seen to behave more erratically.

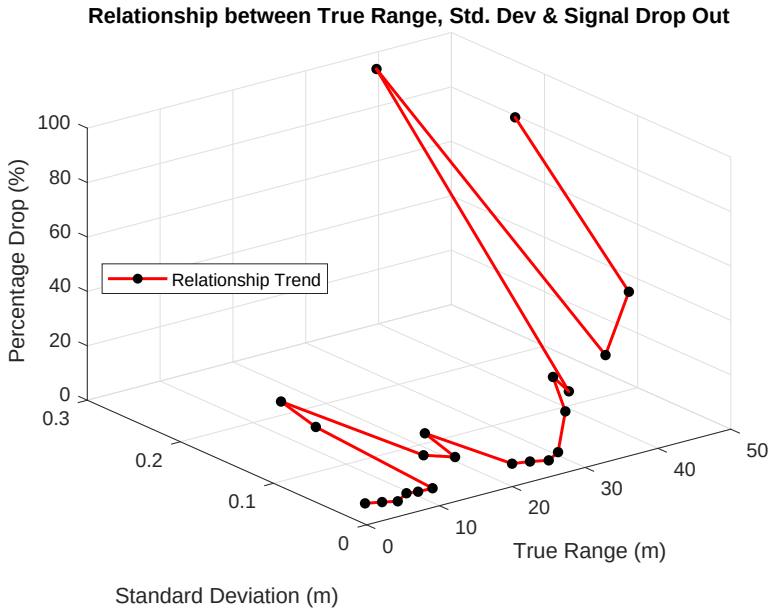


Figure 4.24: 3D relationship between true range, Std. Dev. & percentage of signal dropout

Static Experimental Procedures Conclusion

The completion of static based experimental procedures provided an initial understanding into how the range error produced from two UWB devices behaved in relation to an increase in the distance between them. Such experimental procedures enabled a foundational simulation of the performance of the system when deployed in scenarios where the distance

between two devices would change over time. The UWB system produced an average range error of approximately 0.2m but experienced a significant number of lapses in communication at distances over 30m. However, as mentioned above, the design of sensor fusion algorithms requires the quantification of system uncertainty, which is defined as how much the error deviates from the mean. Although the results above provide foundation understandings of the behaviour of the UWB system error, it was deemed necessary to design an experimental procedure which focused on the study of the standard deviation of error in scenarios that adequately simulated intended dynamic deployment operations. This is covered in the next section.

4.4.2.2 Dynamic Testing

The UWB system was also tested subjected to dynamic testing scenarios. This saw the UWB mounted upon the mobile robot and set to execute both linear and non-linear trajectories whilst simultaneously being tracked by the RTS for ground truth comparison. A dynamic testing environment was necessary to be able to understand how the error of the UWB system behaves during intended mobile operational scenarios. The methodology undertaken and the acquired results are presented in the following sections.

The following methodology, formulation and results are presented in the *International Journal of Sensors MDPI, 2018,* titled *Uncertainty Characterisation of Mobile Robot Localisation Techniques using Optical Surveying Grade Instruments* [32].

Methodological Approach

The robot was set to execute a number of experimental trajectories, where timeseries objects within the MATLAB environment are used to time-synchronise all measurement readings from the RTS and UWB. It is vital that both sources of data are time-synchronised to provide the ability to calculate a point to point comparison along each trajectory. This initially was identified as a main issue as the RTS was static and streaming data into the ground station computer, whereas the UWB system was channelling data to the Raspberry Pi (R-Pi) embedded device on-board the robot. There was no relationship between the robot and the ground station. Due to the R-Pi not possessing a real-time clock (RTC), when the R-Pi is powered down for a short while the internal clock begins to drift and becomes skewed in relation to the global time. An approach designed to overcome this

issue was the implementation of a network time protocol (NTP) client, where the ground station was configured to enable the R-Pi to access its system on boot and acquire the current time of the ground station. This was achieved through the use of a wireless access point (WAP) which was situated within the testing environment. Once this was complete, ROS [57], previously mentioned in Chapter 3 as the on-board system architecture and hub for data transmission and message control, was used to record time-synchronised RTS and UWB measurements. ROS has the ability to implement a master-slave operation, where if ROS is operating on two machines, the data transfer between both machines is transparent enabling unrestricted data sharing. Therefore, with the ground station set as the master and the R-Pi as the slave, the ground station can receive time-synchronised and on-line data transmissions from the mobile robot as it navigates the environment. The whole time-synchronisation procedure resulted in the ability to complete a point to point comparison between the UWB and the RTS for range error calculation. An overview of the system architecture for the UWB dynamic testing is shown in Figure 4.25.

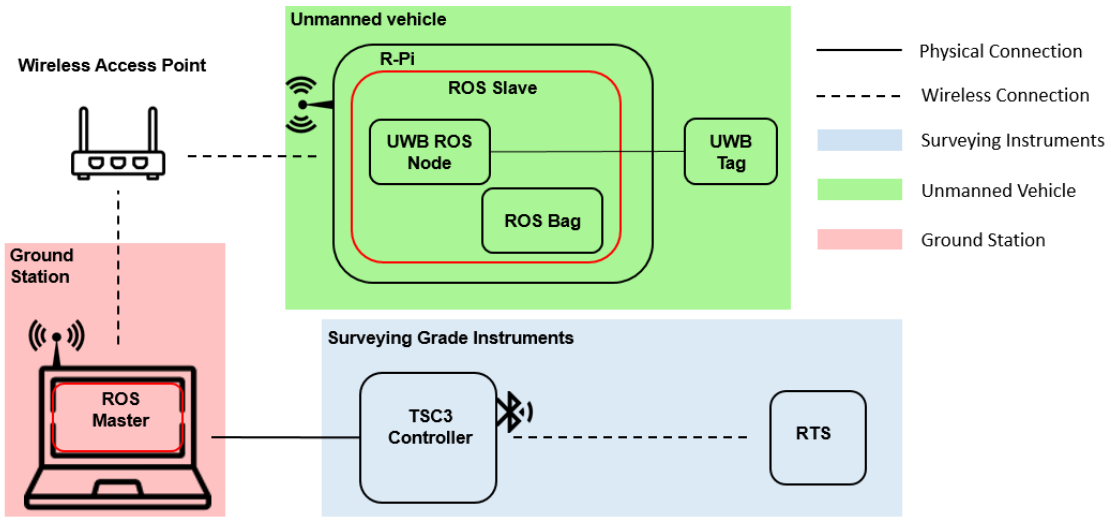
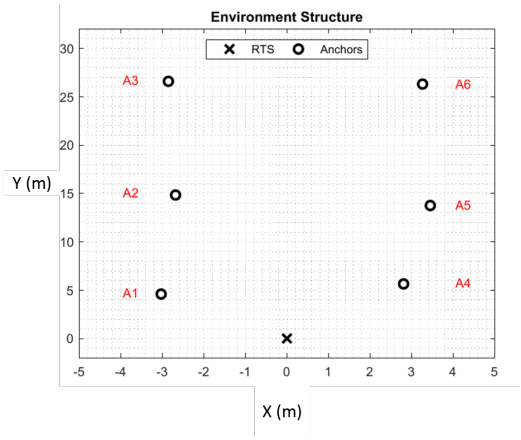


Figure 4.25: Dynamic testing system architecture

An indoor and uncluttered testing environment was designed which saw the RTS configured to the identical set-up discussed at the beginning of this chapter. The RTS was set to a calibrated known location within its local coordinate frame. This therefore meant that all measurements to the robot acquired from the RTS were referenced from the origin of the RTSs' local position, not within a geo-referenced coordinate frame as the indoor environment restricted the ability to geo- reference this testing procedure. An

indoor and uncluttered environment was chosen initially to analyse the behaviour of the UWB positioning system's native algorithm and the resulting range error produced in a simplistic environment. An image of the testing environment is shown in Figure 4.26a. An important section of this testing procedure was how the precise locations of each anchor were acquired and provided to the UWB ROS node on-board. The RTS was configured to take a static shot to each UWB anchor antenna, where the UWB signal originates from, to acquire its true location with millimetre precision. This acquisition procedure of true anchor locations is shown in Figure 4.27, where the image is taken directly from the theodolite's telescope and the resulting anchor set-up structure is demonstrated in Figure 4.26b.

(a)

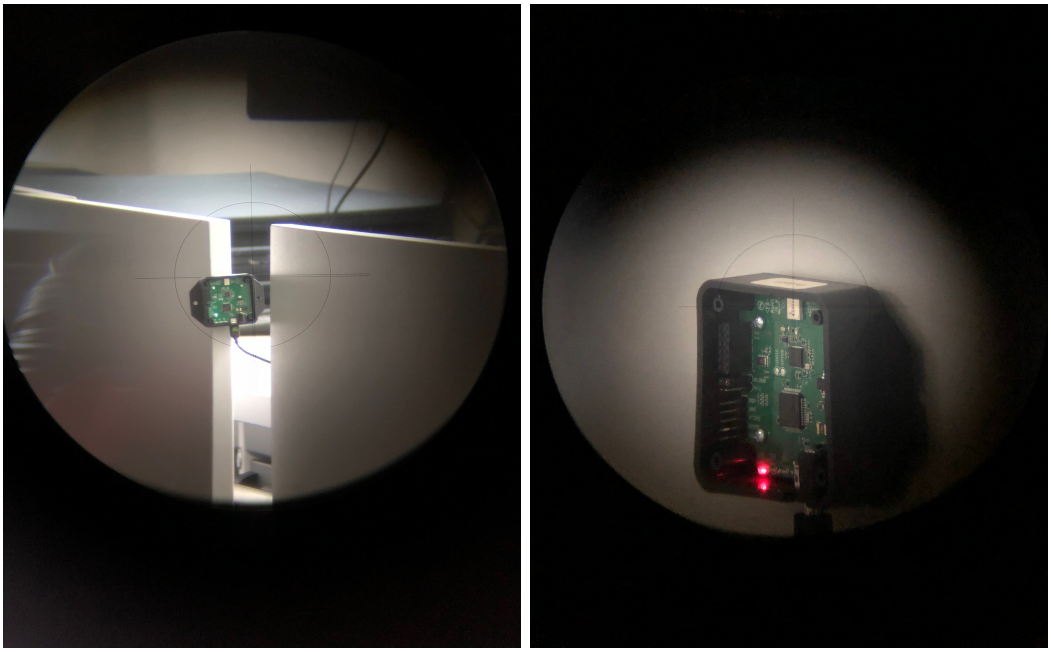


(b)

Figure 4.26: Environment Structure (a) Indoor open testing environment (b) Anchor Node Locations

The locations of each anchor within the environment are presented in Table 4.5.

Post configuration of the static anchors, the robot was then set to execute two types of raster patterns with an attempt to cover all of the environment enclosed by the anchors, with each raster pattern conducted at two repetitions. The raster patterns are shown in Figure 4.28 and the details concerning repetitions and measurement quantities are presented in Table 4.6. The two conducted raster patterns were executed both vertically and horizontally to assess the UWB systems measurement variance in both the X and Y axes as shown in Figure 4.28, additionally, abbreviations SP and EP denote start point and end points respectively.



(a) (b)

Figure 4.27: Direct static shot to measure true location of UWB anchors

Table 4.5: Locations of RTS and UWB Anchors referred to as nodes

Node	**X (m)**	**Y (m)**
RTS	0.00	0.00
A1	-3.0270	4.591
A2	-2.681	14.833
A3	-2.853	26.581
A4	3.263	26.2970
A5	3.449	13.745
A6	2.807	5.639

Figure 4.29 shows the resulting trajectory estimation from the UWB system's native localisation algorithm in comparison to the RTS. It is evident that for the conducted testing raster patterns both horizontally and vertically orientated, the UWB system's native localisation algorithm generated considerable noisy position estimations, especially for the first two raster patterns. The position estimations from the UWB system's native algorithm stem from individual range measurements from each anchor, therefore stating that the position estimation is only as good as the provided range measurements from each anchor. Based on the UWB output shown in Figure 4.29, it is evident that there is a clear need for a method of improving the system's overall position estimations. This begins

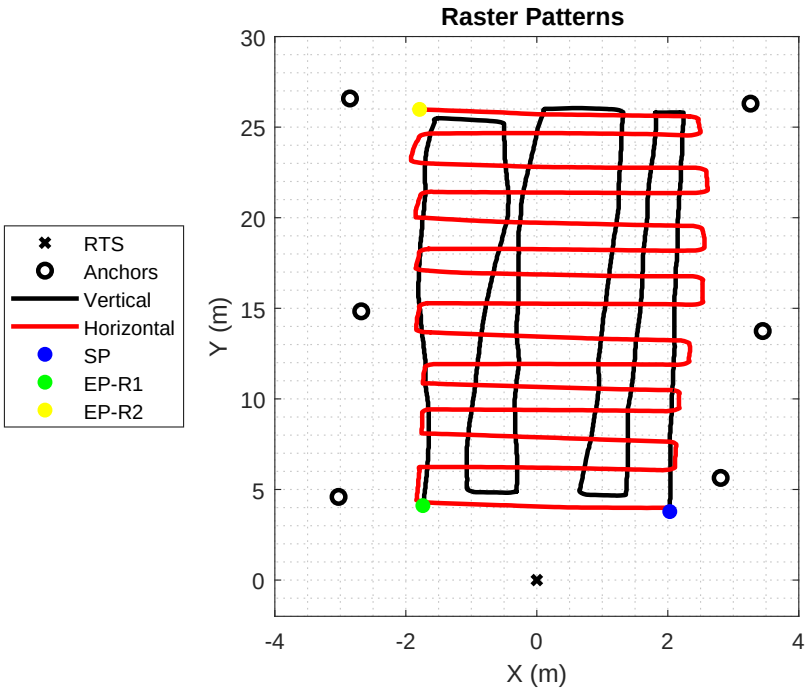


Figure 4.28: Raster Patterns for measurement acquisition

Table 4.6: Overview of Dynamic test structure

Traj.	**Repetitions**	**Total No. Measurements**
Vertical Raster	2	8500
Horizontal Raster	2	7500

with characterising the anchor range measurement variance and then integrating this with an implementation of a sensor fusion algorithm to improve overall position estimations. However, as the measurements acquired from the experimental raster patterns will be used to quantify such range error, due to the extreme outliers shown Figure 4.29, a data cleaning process is required. This process is introduced in the following section prior to the quantification of UWB range error.

Data Cleaning

The data cleaning process was conducted to remove outliers that could eventually have an effect when quantifying the variance of the UWB range measurements. It was evident from observing the UWB system's native algorithm that there were some outlier position estimations which were calculated from erroneous range measurements. The initial stage was to isolate and observe both the X and Y components to identify which axis the outlier

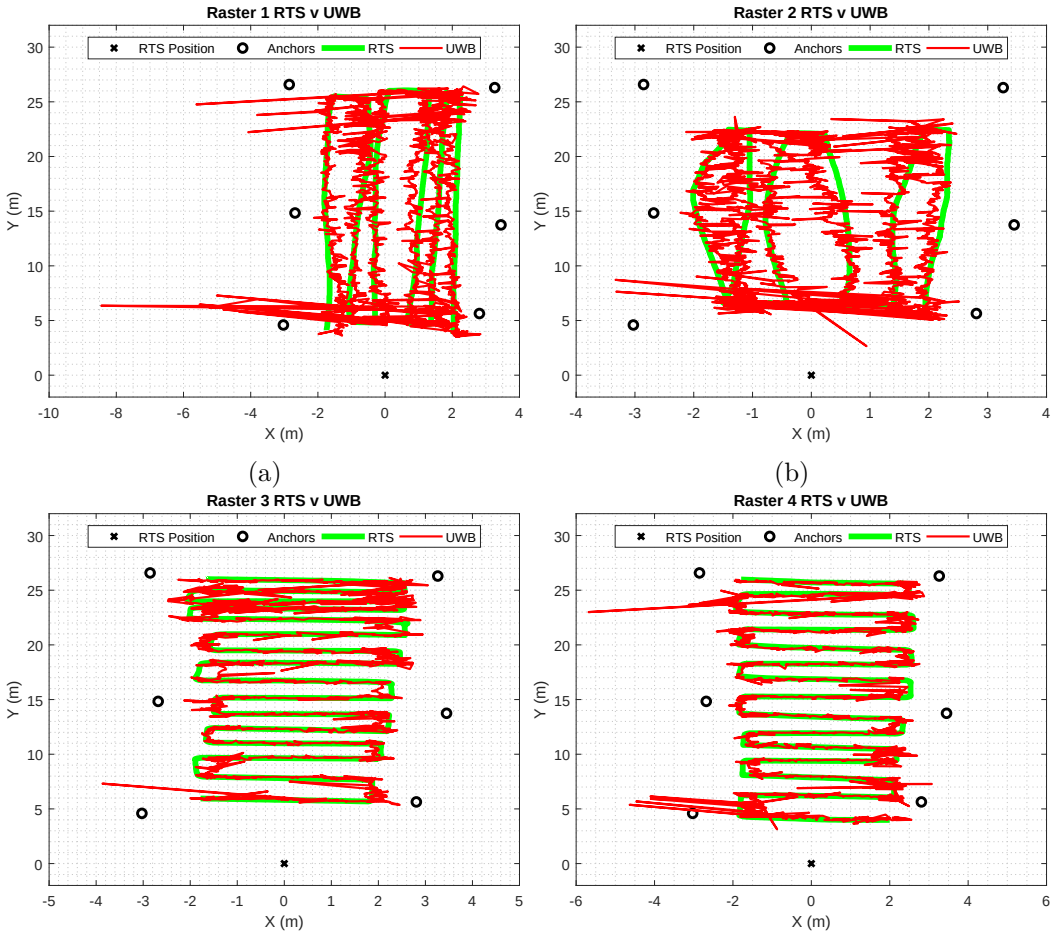


Figure 4.29: RTS v UWB for position estimation (a) Raster Pattern 1 (b) Raster Pattern 2 (c) Raster Pattern 3 (d) Raster Pattern 4

estimations occurred on. For each of the experimental raster patterns, the X and Y axes results compared to those of the RTS can be observed in Figure 4.30 and Figure 4.31 respectively.

It is evident from the observation of both Figure 4.30 and Figure 4.31 that outlier Cartesian position estimations were a cause of erroneous measurements on the X axis as the is a lot more noise present compared to that on the Y axis. Therefore, the results from the X axis were used as a means to remove outliers. The outlier identification and removal technique was conducted through the use of the Z-score. The Z-score is a statistical technique which identifies the number of standard deviations that the data point lies from the mean. With the use of a threshold, outlier points can be identified and removed. The formula used to calculate the Z-score of each data point is shown in Equation (4.7).

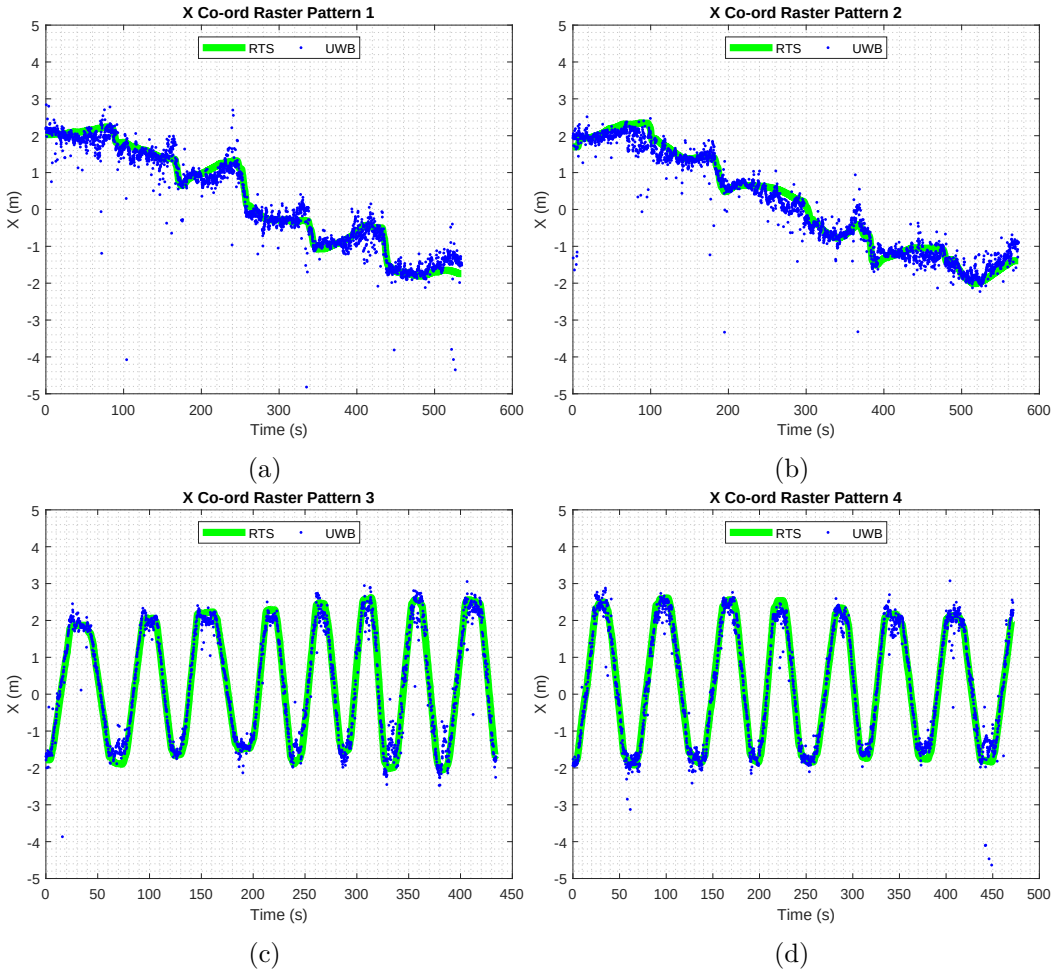


Figure 4.30: RTS v UWB for X position estimation (a) Raster Pattern 1 (b) Raster Pattern 2 (c) Raster Pattern 3 (d) Raster Pattern 4

$$z_i = \frac{x_i - \bar{\mu}}{\sigma} \tag{4.7}$$

Where;

- z_i - Z score of data point i
- x_i - Data point i
- $\bar{\mu}$ - Dataset mean
- σ - Dataset standard deviation

The process executed to identify outliers was to apply the Z-score technique to the X position estimation error acquired from the UWB system's native algorithm. This error was calculated through Equation (4.8) for each of the four raster patterns.

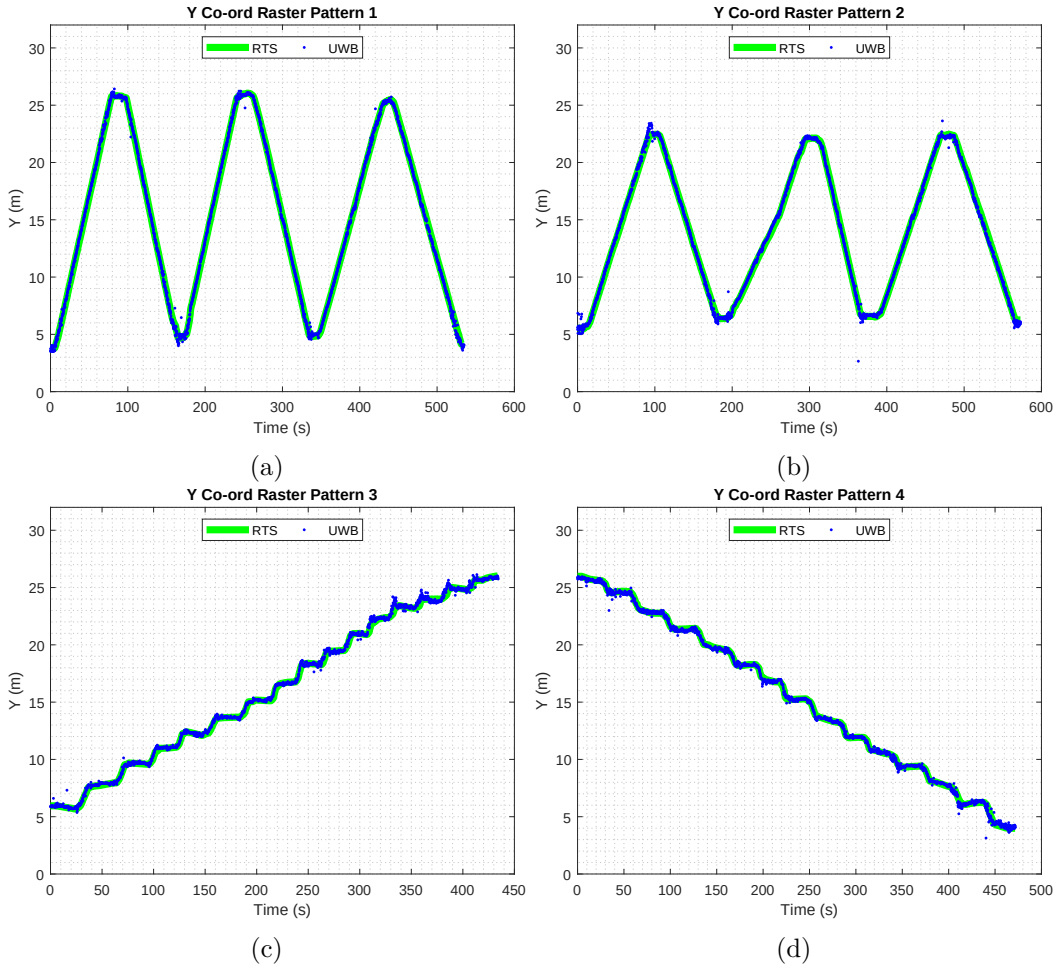


Figure 4.31: RTS v UWB for Y position estimation (a) Raster Pattern 1 (b) Raster Pattern 2 (c) Raster Pattern 3 (d) Raster Pattern 4

$$\epsilon_{X_i} = RTS_{X_i} - UWB_{X_i} \tag{4.8}$$

Where;

- ϵ_{X_i} - Error of X position at data point i
- RTS_{X_i} - RTS X position estimate at data point i
- UWB_{X_i} - UWB X position estimate at data point i

Post calculation of ϵ_X for each raster pattern, the Z-score of each of the four X position error datasets were calculated and a threshold of $-3 < z < 3$ standard deviations was set to identify and remove extreme outliers. Figure 4.32 represents the resulting Z scores obtained for each raster pattern concerning the X position error. It shows which readings acquired are in-fact extreme outliers and identified as the large red markers and subsequently removes

them prior to quantifying the UWB range error.

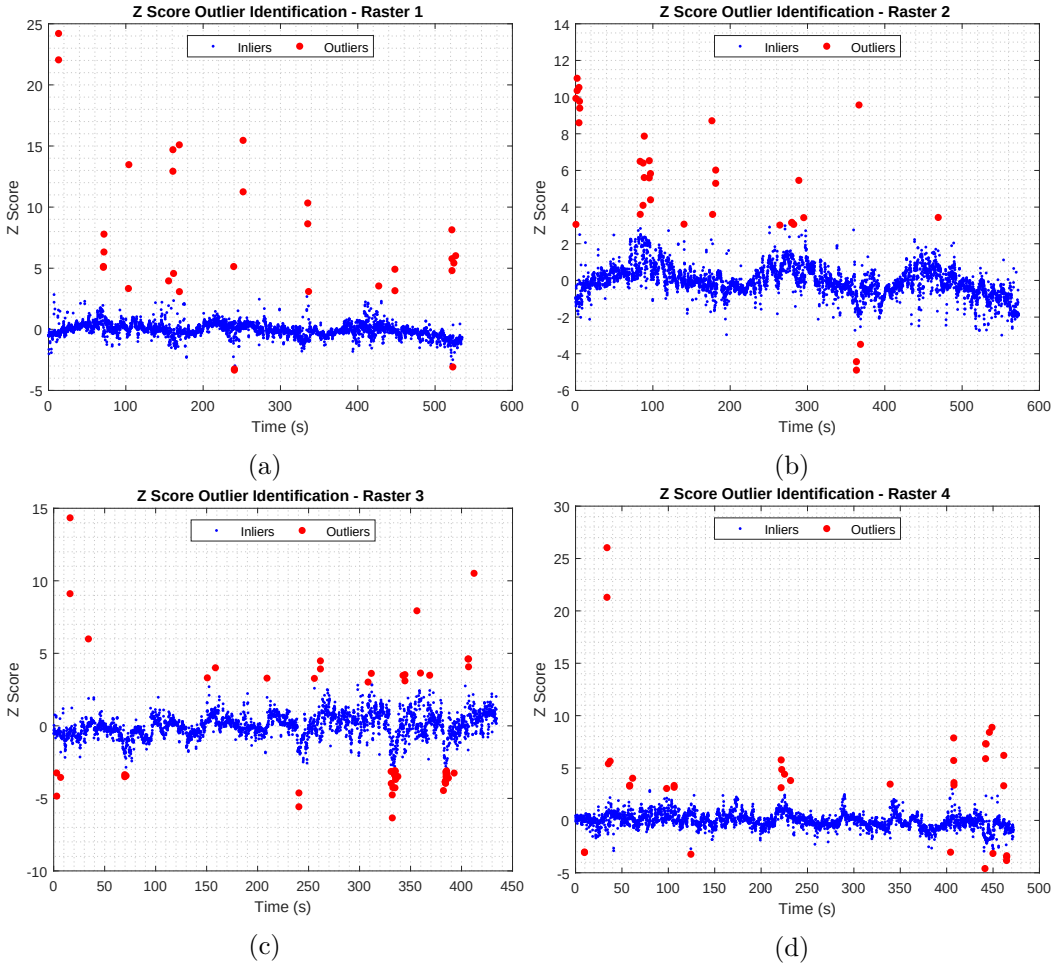


Figure 4.32: Z Score for X position Error (a) Raster Pattern 1 (b) Raster Pattern 2 (c) Raster Pattern 3 (d) Raster Pattern 4

Calculating Range Error

Using the RTS to take static direct shots to the UWB anchor, specifically to the antenna where the signal providing the range measurement originates from, enabled the acquisition of the true location of each anchor in relation to the RTS position. As the robot manoeuvres through the environment, each update acquired from the UWB system's native localisation algorithm consists of six raw range measurements acquired from each individual anchor mapped as $r_k = [r_{Ai}, i \in [1, ..., 6]]$ at time instance k, where r is the range measurement acquired from anchor A_i. However, as the RTS is also providing a position of the robot through the interaction with the on-board optical prism, it was possible to calculate a set of true ranges r_{t_k} also mapped as $r_k = [r_{t_{Ai}}, i \in [1, ..., 6]]$ at time instance k, where $r_{t_{Ai}}$ is the true range measurement from anchor A_i to the robot. Equation (4.9) shows the

calculation executed to acquire the true range or Euclidean distance from anchor A_i to the robot using the robot's location update at time instance k acquired from the RTS and the known location of anchor A_i.

$$r_{t_{A_ik}} = \sqrt{(x_{t_{Rk}} - x_{A_i})^2 + (y_{t_{Rk}} - y_{A_i})^2} \tag{4.9}$$

Where;

- $r_{t_{A_ik}}$ - True range between the robot and anchor A_i at time k
- $x_{t_{Rk}}$ - True Cartesian x component of the location of the robot at time k
- x_{A_i} - Cartesian x component of the location of anchor A_i
- $y_{t_{Rk}}$ - True Cartesian y component of the location of the robot at time k
- y_{A_i} - Cartesian y component of the location of anchor A_i

Therefore, range error was calculated through Equation (4.10).

$$\epsilon_{A_ik} = r_{t_{A_ik}} - r_{A_ik} \tag{4.10}$$

Where;

- e_{A_ik} - Range error acquired from anchor A_i at time k
- $r_{t_{A_ik}}$ - True range between the robot and anchor A_i at time k
- $r_{t_{A_ik}}$ - Range measurement acquired from anchor A_i at time k

Therefore, using Equation (4.9) and Equation (4.10) enabled the quantification of the range error of each UWB anchor for the four raster patterns. However, as previously discussed, the removal of extreme outliers that may have an everlasting effect on the range error metrics enabled the calculation of the error metrics. The acquired results are presented in the following section.

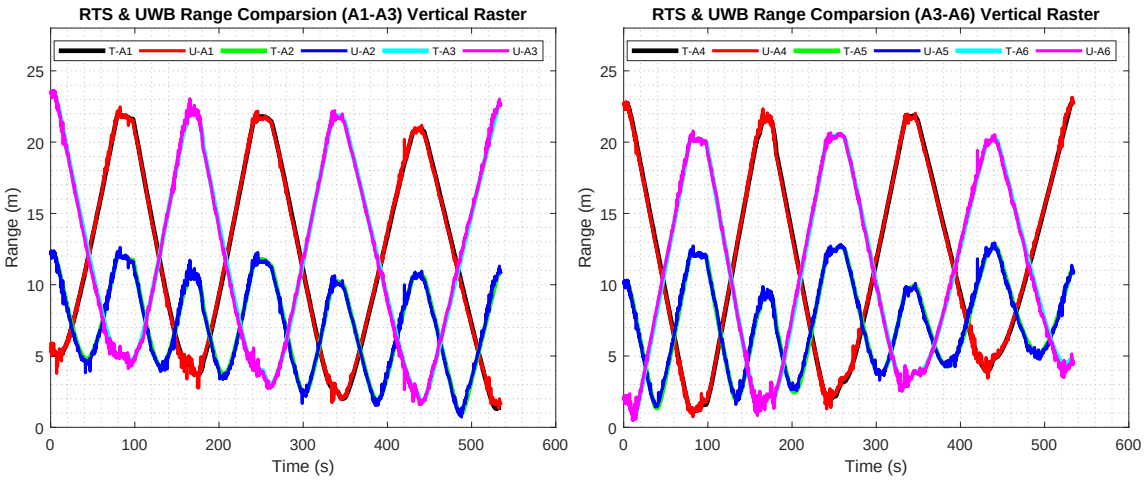
Results

The implementation of Equation (4.9), Equation (4.10) and the integration of the raw range measurements acquired from the UWB system enabled an estimated versus true range

comparison for each trajectory. Firstly, the true ranges from the RTS are compared against each individual anchor for both the horizontal and vertical raster patterns to demonstrate the nature of such range measurements. Secondly, all four raster patterns both vertically and horizontally orientated are combined to form one individual dataset. It is here where the range error is quantified.

Vertical Raster Patterns

The range results gathered from the vertical raster patterns (raster patterns 1 & 2) are shown in Figure 4.33, where denotations T and U included within the legends represent true measurements acquired from the RTS and native UWB measurements respectively. Figure 4.33 shows that the raw UWB range measurements to the robot from all six anchors generally follow a similar trend to the ground truth readings acquired from the RTS. However, it is obvious that the UWB measurements increase in volatility in areas where the range to an anchor is seen to experience a sudden change. This can be identified especially when observing Anchors 1, 3, 4 and 6. These areas of the raster pattern where the range is seen to suddenly change direction may be due to the unmanned system executing a turning procedure. Further showing that the range measurements become volatile as reduces it velocity and its rate of movement.

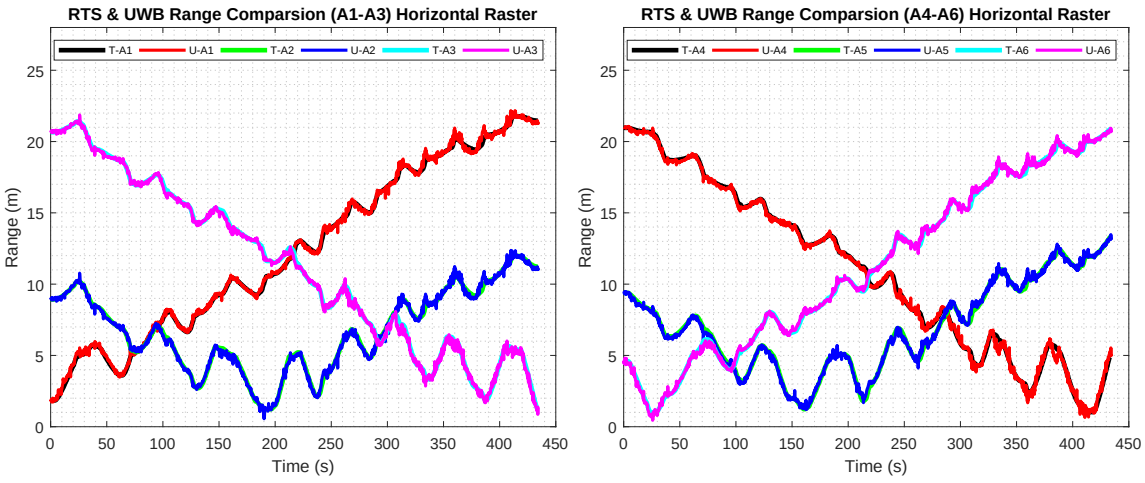


(a) (b)

Figure 4.33: Anchor Range vs True Range for Vertical Rasters (a) Anchors 1-3 (b) Anchors 4-6

Horizontal Raster Patterns

The results gathered from the horizontal raster patterns are shown in Figure 4.34. These results show that the error behaviours are similar to those shown in Figure 4.33 for the vertical raster patterns, where the UWB range generally follows a similar trend to the ground truth of the RTS. However, the underlying differences between the error results generated for horizontal and vertical raster patterns is related to nature of the conducted trajectory. The range observations shown in Figure 4.34 demonstrate slightly more noisy range measurements in comparison to those acquired during the vertical raster patterns. This can be observed in areas where the range to an anchor increases linearly, as for the horizontal raster pattern results, volatility is observed within these segments. This, however, is not the case for the previously presented range measurement for the vertical raster patterns.



(a) (b)

Figure 4.34: Anchor Range vs True Range for Horizontal Rasters (a) Anchors 1-3 (b) Anchors 4-6

Raster Combination

It was clear from the conducted raster patterns presented in the previous sections that the robot experienced a good coverage of the environment enclosed by the anchors, therefore alternating the range to each anchor to assess the behaviour of the range error metric. As a result, the error metrics acquired from all four raster patterns were combined to calculate two error metrics. These error metrics are outlined below;

- Generalised - A generalised error metric for the whole UWB system. This included

the combination of all anchor errors acquired over the four raster patterns.

- Specific - A specific error metric is defined as an error metric specific to each anchor. Instead of totalling errors from all present anchors and taking a generalised metric, this error metric results in each anchor having its own individual error metric.

The necessity to acquire two different types of metric metrics for the UWB system was to observe how the designed R-EKF performed with two different techniques for the measurement covariance matrix R. Therefore, in order to meet assumptions made by the EKF, the range error values acquired were modelled as normal distribution calculated through Equation (4.11). The reasoning behind this modelling approach is that when designing an EKF, the measurement noise R element of the algorithm is modelled as a zero mean Gaussian distribution satisfying the statements shown in Equations (4.12) to (4.14).

$$y = f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{4.11}$$

$$R = \mathcal{N}(\mu, \sigma^2) \tag{4.12}$$

$$\mu \in \mathbb{R} \tag{4.13}$$

$$\sigma^2 > 0 \tag{4.14}$$

Where;

- μ - Mean error
- σ - Standard deviation of error
- σ^2 - Error variance

The combined error statistics for each individual anchor are presented in Figure 4.36. Figure 4.36 as a whole shows that all anchors experienced similar maximum standard deviation or error, however, anchors 1, 2, 3, 4 and 6 shown in Figure 4.35a, Figure 4.35b, Figure 4.35c, Figure 4.35d and Figure 4.35f respectively demonstrate a more constructed

Gaussian distribution of error. This is identified by focusing on how the shape of the histogram bins representing error frequency follows that of the bell curve. Contrastingly, anchor 5 shown in Figure 4.35e demonstrates a less constructed Gaussian distribution. It is observed that all anchors have a similar mean error averaging at approximately zero. The mean error metric can be identified in the normal distribution plot as the corresponding X axis value relating to the peak of the bell curve, signifying the highest probability. From each of the histogram plots shown in Section 4.4.2.2, it is possible to extract the error statistics required to construct the specific error metric.

With the range error results generated from the isolated assessment of each individual anchor used as a means to acquire the specific error metrics, a final compilation of measurements was constructed to provide a generalised error metric, combining data from all anchors. A normal distribution assessment was also executed on the final dataset, this is shown in Figure 4.36. Figure 4.36 shows that the generalised range error of all UWB anchors has a Gaussian distribution with a mean located at approximately zero metres. Identifying the distribution behaviour of the error matches the requirements of how an EKF is formulated, stating that the measurement noise variable satisfies Equations (4.12) to (4.14).

It is also interesting to observe that this zero mean and RMSE concerning the UWB range for the generalised errors metric is generally consistent as the true range increases. This is observable in Figure 4.37, notably by the heatmap presented in Figure 4.37b. The presentation of the heatmap provides an overview of the relationship between the range error and the increasing true range.

The final error statics for both the specific and generalised methods are shown in Table 4.7, where each of the anchors is seen to have similar RMSE in the range of 0.17m - 0.21m and near zero mean. The generalised metrics also show that the UWB system as a whole has a slight bias, even though Figure 4.36 visually demonstrates a near zero mean of error, the UWB system possesses a small bias of 8.5mm. The acquisition of such metrics were carried into the design of the R-EKF.

Dynamic Testing Conclusions

To conclude the UWB dynamic testing procedure, the range estimations produced from the UWB were generally similar to the ground truth of the RTS for an open and un-cluttered

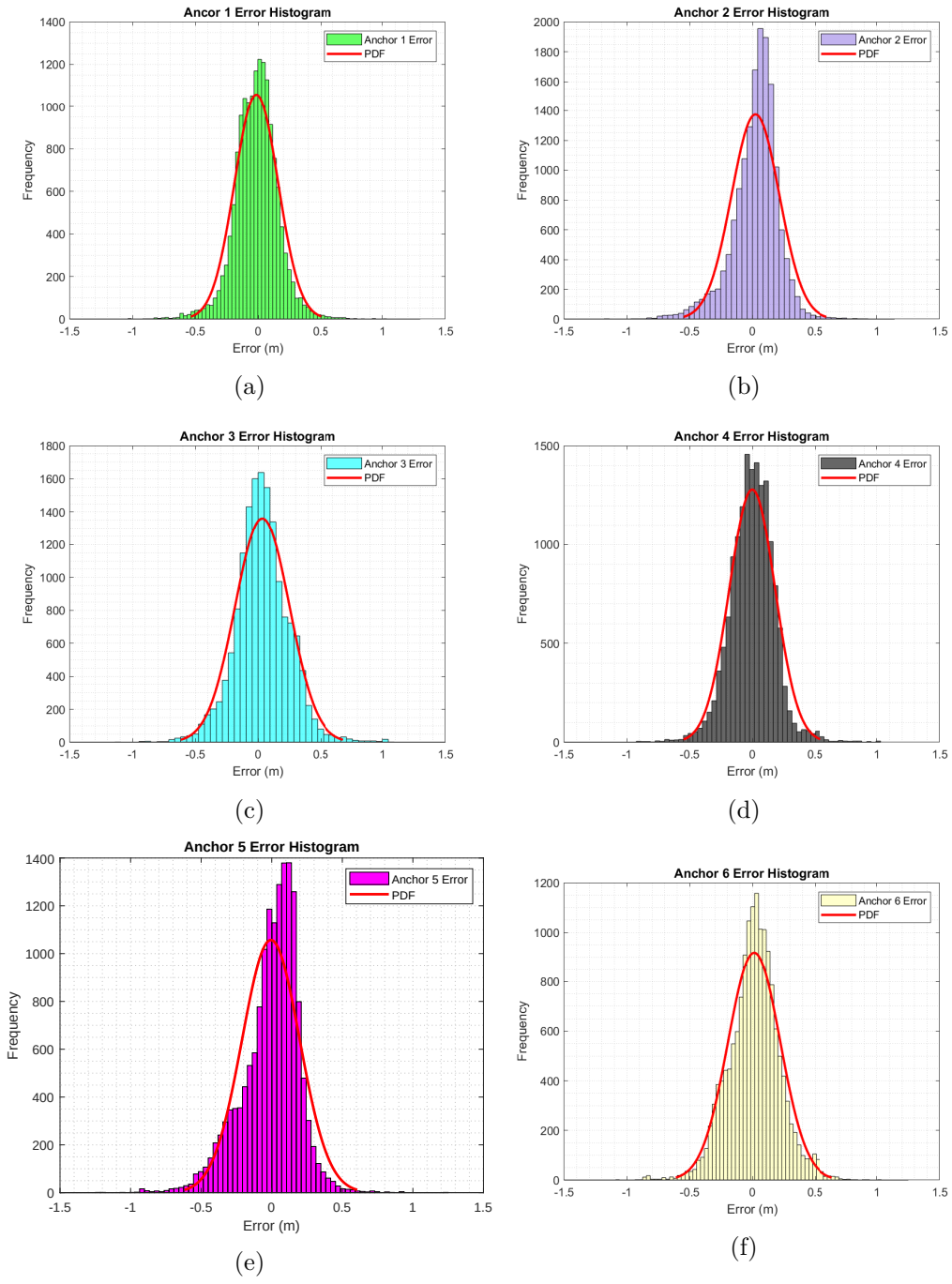


Figure 4.35: Distributions of Error (a) Anchor 1 (b) Anchor 2 (c) Anchor 3 (d) Anchor 4 (e) Anchor 5 (f) Anchor 6

indoor environment. However, sudden discontinuous changes in range to an anchor effected the accuracy of the range estimation as fluctuations in error were seen to occur. As a whole, when the error for all anchors were compiled to acquire a generalised error statistic, the range error produced a Gaussian distribution of error with a mean of 8.5mm demonstrating a resulting low-scaled error bias. Additionally, the generalised RMSE was calculated at

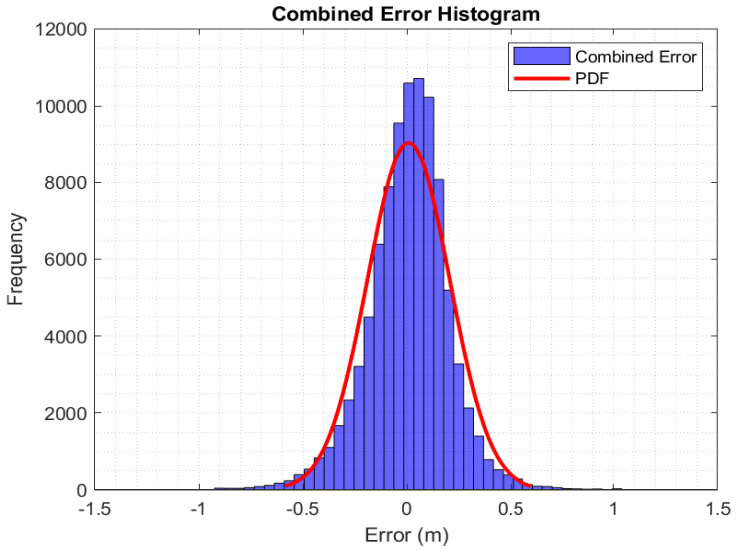


Figure 4.36: Distribution of Error from all Anchors

Table 4.7: Range error statistics for each anchor

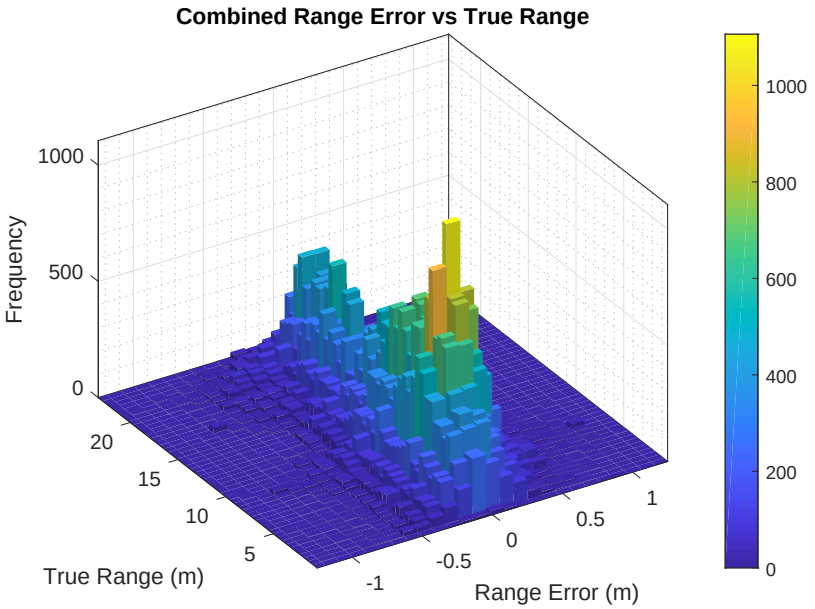
Anchor No.	μ **(m)**	σ **(m)**
1	-0.0139	0.1738
2	0.0244	0.1907
3	0.0336	0.2167
4	-0.0013	0.1824
5	-0.0061	0.2039
6	0.01423	0.2082
Generalised	0.0085	0.1972

20.82cm, where all six anchors produced similar outputs. Additionally, each anchor held its own specific set of error statistics that were also recorded. These calculated error statistics were then taken and integrated into the construction of the R-EKF. Therefore, the range measurement error bias and standard deviation for the UWB that will be integrated into the R-EKF are presented in the two following subsections for both the generalised and specific methods. Additionally, the observation covariance matrix denoted as R which will be utilised in then R-EKF for both methods is also included.

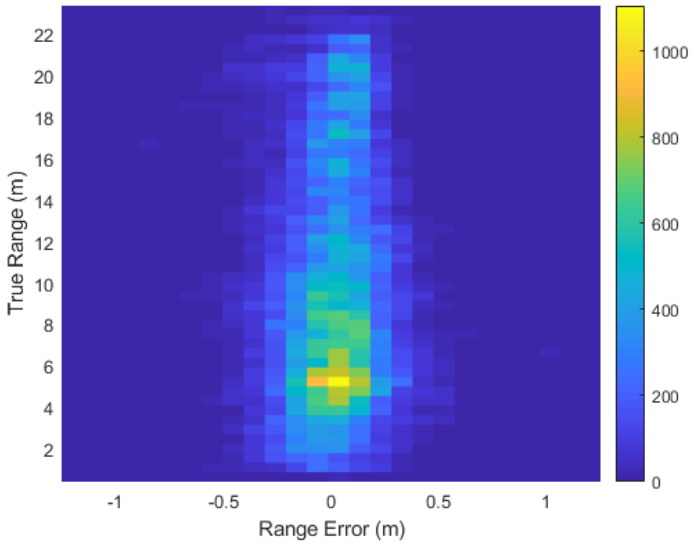
Generalised

$$\beta_{r_{A_G}} = 0.0085\text{m} \tag{4.15}$$

$$\sigma_{r_{A_G}} = 0.2082\text{m} \tag{4.16}$$



(a)



(b)

Figure 4.37: Combined Error vs True Range (a) 3D Histogram (b) Heatmap

$$R_G = \begin{bmatrix} \sigma^2_{r_{A_G}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{r_{A_G}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{r_{A_G}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{r_{A_G}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{r_{A_G}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{r_{A_G}} \end{bmatrix} \tag{4.17}$$

Specific

$$\beta_{r_{A_{[1\ldots6]}}} = [-0.0139,\ 0.0244,\ 0.0336,\ -0.0013,\ -0.0061,\ 0.01423]\text{m} \tag{4.18}$$

$$\sigma^2_{r_{S_{A_{[1\ldots6]}}}} = [0.1738,\ 0.1907,\ 0.2167,\ 0.1824,\ 0.2039,\ 0.2082]\text{m} \tag{4.19}$$

$$R_S = \begin{bmatrix} \sigma^2_{r_{A1_S}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{r_{A2_S}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{r_{A3_S}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{r_{A4_S}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{r_{A5_S}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{r_{A6_S}} \end{bmatrix} \tag{4.20}$$

4.5 Range based EKF Localisation Formulation

This section introduces the formulation of the components of the R-EKF. It takes the previously calculated uncertainty metrics for the gyroscope and wheel encoders which are utilised as a control input into the motion model and the observation covariance matrix R which is included in the observation motion. Prior to the introduction into the R-EKF, for reference purposes it is observed that estimated parameters will be denoted with the inclusion of a caret symbol as such $\widehat{\chi_k}$. This is also the case for a priori parameters, which are defined as parameters prior to the application of the observation update, and are denoted with the inclusion of a bar symbol as such $\widehat{\chi_k^-}$. The formulation of the R-EKF is presented in the following sections. Additionally, the formulation of the R-EKF is also included in [32].

4.5.1 Problem Formulation

A discrete EKF was used as the sensor fusion object to combine the initial position estimation from a control input with external range observations updates acquired from the UWB anchors. The EKF calculates a linear approximation for a set of non-linear functions based on the first-order Taylor expansion. This is achieved through a two stage iteration process where the future state of the system is predicted based on the current state

and a motion model, this is then corrected using observations from an external source. The non-linear motion and observation models for a conventional EKF are described in Equation (4.21) and Equation (4.22) where $\widehat{\chi_k}$ and $\widehat{Z_k}$ represent the state and observations vector estimations at time k, controlled input into the system is represented as u_k and notations w_{k-1} and v_{k-1} are the system and observation noise.

$$\widehat{\chi_k} = f(\chi_{k-1}, u_k; w_{k-1}) \tag{4.21}$$

$$\widehat{Z_k} = h(\chi_{k-1}, v_{k-1}) \tag{4.22}$$

For the system implemented in this study, the range measurements acquired from the UWB anchors meant that the system was assessed as a range-based localisation problem. The problem when concerning mobile robots includes the continuous estimation of the robot's 2D state in terms of its planar Cartesian coordinates and heading, in relation to the reference anchors in the environment which are assumed to have static positions mapped in $M_{Bi} = [x_{Bi}\, y_{Bi}], i \in [1 \ldots N_B]$ and provide a set of range measurements $Z_{Bi} = [r_{Bi}, i \in [1 \ldots N_{Bi}]]$ at time k, where N_B is the number of UWB anchors present. Therefore, the state vector for the R-EKF is shown in Equation (4.23).

$$\chi_k = [x_k\ y_k\ \psi_k]^T \tag{4.23}$$

Where;

- x - Cartesian co-ordinate x (m)
- y - Cartesian co-ordinate y (m)
- ψ - Heading (rad)

The R-EKF was used to integrate the initial position estimate with the external range observations acquired from the UWB beacons. For the design of the R-EKF, initial state predictions were based on systematic mathematical modelling fused with a control odometry input, where quadrature encoders are used to determine the angular displacement of the robot's wheels at time k as well as a time-synchronised gyroscope for heading calculation. The control input vector from the definitions of Equation (4.21) and Equation (4.22), the motion and observations models can be defined and formulated. This is presented in the

following sections.

4.5.2 Motion Model

The motion model is a formulation of the mobile robot kinematics and in this case, how the control input measurements into the system temporally transition the components of the state vector χ. The inclusion of a control input source u_k into the motion model enables an improved estimation of how the states transition rather than solely relying on a time-step. With the state vector χ defined, the measurements acquired from the quadrature encoders and gyroscope form the control input vector u for each time iteration k. This is shown in Equation (4.24).

$$u_k = [\theta_{L_k}\ \theta_{R_k}\ \delta\psi_k]^T \tag{4.24}$$

Where;

- θ_{L_k} - Left wheel angular displacement (rad)
- θ_{R_k} - Right wheel angular displacement (rad)
- $\delta\psi_k$ - Heading angular displacement (rad)

Therefore, the motion model of the system is modelled based on the kinematics of a system with a differential drive (DD) steering schematic. It has previously been mentioned in Chapter 3 that the mobile system has in fact been designed with a skid steering schematic. The skid steering schematic experiences wheel slippage when turning or changing heading. However, if the occurrence of wheel slippage is neglected, the skid-steering schematic can be modelled as a DD. The result of this assumption consequently incorporates inaccuracies into the motion model which are therefore required to be integrated into the formulation of the R-EKF. The motion model is shown in Equation (4.25) and Equation (4.26) [12].

$$\widehat{\chi}_k^- = f(\widehat{\chi}_{k-1}, u_k) \tag{4.25}$$

$$f(\widehat{\chi}_{k-1}, u_k) = \widehat{\chi}_{k-1} + \begin{bmatrix} \delta D_k \cos(\psi_k + \frac{\delta\psi_k}{2}) \\ \delta D_k \sin(\psi_k + \frac{\delta\psi_k}{2}) \\ \psi_k + \delta\psi_k \end{bmatrix} \tag{4.26}$$

Where δD_k is the linear displacement at time k and is calculated using the radii of the wheels r_L, r_R, as well as the change in angular displacement θ_{L_k} and θ_{R_k} acquired from the pulses of the encoders located in the control input vector u. The calculation of δD_k is shown in Equation (4.27).

$$\delta D_k = \frac{r_L \delta\theta_{L_k} + r_R \delta\theta_{R_k}}{2} \tag{4.27}$$

The theory behind the conventional EKF and why it is used rather than standard Kalman Filters is how it deals with the non-linearity of the system. As seen in Equation (4.26), it is evident that the motion model of the mobile robot is non-linear, therefore justifying the necessity of the implementation of non-linear state estimators such as the EKF. The EKF approximates the linearisation of a non-linear system through the execution of the first-order Taylor expansion represented as Jacobian matrices G_x and G_u. Where, G_x is defined as the Jacobian matrices of the motion model with respect to the state vector and G_u is the Jacobian of the motion model with respect to the control input vector. The conventional terms for both Jacobian matrices are shown in Equation (4.28) and Equation (4.29).

$$G_{x_k} = \frac{\partial f}{\partial x}(\widehat{\chi}_k^-, u_k) \tag{4.28}$$

$$G_{u_k} = \frac{\partial f}{\partial u}(\widehat{\chi}_k^-, u_k) \tag{4.29}$$

Therefore, the formulation of G_x and G_u was achieved through the utilisation of the MATLAB symbolic toolbox which enables the non-numerical formulation of an equation prior to actual implementation. The formulations for G_x and G_u are presented in Equation (4.30) and Equation (4.31) respectively.

$$G_{x_k} = \begin{bmatrix} 1 & 0 & -\frac{\delta D_k \sin\left(\psi_k + \frac{\delta\psi_k}{2}\right)}{2} \\ 0 & 1 & \frac{\delta D_k \cos\left(\psi_k + \frac{\delta\psi_k}{2}\right)}{2} \\ 0 & 0 & 1 \end{bmatrix} \tag{4.30}$$

$$G_{u_k} = \begin{bmatrix} \frac{r_L \cos\left(\psi_k + \frac{\delta\psi_k}{2}\right)}{2} & \frac{r_R \cos\left(\psi_k + \frac{\delta\psi_k}{2}\right)}{2} & -\frac{D_k \sin\left(\psi_k + \frac{\delta\psi_k}{2}\right)}{2} \\ \frac{r_L \sin\left(\psi_k + \frac{\delta\psi_k}{2}\right)}{2} & \frac{r_R \sin\left(\psi_k + \frac{\delta\psi_k}{2}\right)}{2} & \frac{D_k \cos\left(\psi_k + \frac{\delta\psi_k}{2}\right)}{2} \\ 0 & 0 & 1 \end{bmatrix} \tag{4.31}$$

The standard deviation for both sources of the control input were then integrated into the design of the control input covariance matrix U. The formulation of U is shown in Equation (4.32).

$$U = \begin{bmatrix} \sigma^2_{\theta_L} & 0 & 0 \\ 0 & \sigma^2_{\theta_R} & 0 \\ 0 & 0 & \sigma^2_{\delta\psi} \end{bmatrix} \tag{4.32}$$

Where;

- $\sigma_{\theta_L} = 0.2$ deg
- $\sigma_{\theta_R} = 0.2$ deg
- $\sigma_{\delta\psi} = 1.199$mdeg/s

With the motion model being dependent upon the control input from the encoders and gyroscope, the process noise Q is constructed using the control input covariance U. This means that the inaccuracies due to the modelling of the process will be the result of the uncertainty of the control input. This is shown in Equation (4.33).

$$Q = G_{u_k}\ U\ G^T_{u_k} \tag{4.33}$$

4.5.3 Observation Model

The observed sensor measurements collected from UWB devices are range measurements representing the Euclidean distances between the static reference anchors A_{B_i} and the estimated position of the robot. At each measurement update, six anchors each provide a range measurement $Z_{Bi} = [z_1, z_2 ... z_6]$, which represents the observation vector that is used to complete the update step of the algorithm. It is noted that the difference between Z_{Bi} and $\widehat{Z}_{Bi}$ is that the Z_{Bi} is the vector of external observations received by the UWB anchors, whereas $\widehat{Z}_{Bi}$ is the predicted estimated observations generated from the observation function h. The process of the observation function is to convert the a priori estimated state $\widehat{\chi}_k^-$ into the equivalent format of Z_{Bi}, this is conducted to enable the calculation of the observation residual Y. The general form of the observation function is shown in Equation (4.34).

$$\widehat{Z}_{Bi} = h(\widehat{\chi}_k^-, A_{B_i}) \tag{4.34}$$

The formulation generated using the MATLAB symbolic toolbox is then shown in Equation (4.35), where it is evident that the formula implemented follows the equation of a circle law taking the location of the anchor as the centre and utilising the Pythagoras theorem to calculate the predicted range observation estimations.

$$\widehat{Z}_{Bi} = h(\widehat{\chi}_k^-, A_B) = \begin{bmatrix} \sqrt{(x_k - \mathrm{x}_{A_1})^2 + (y_k - \mathrm{y}_{A_1})^2} \\ \sqrt{(x_k - \mathrm{x}_{A_2})^2 + (y_k - \mathrm{y}_{A_2})^2} \\ \sqrt{(x_k - \mathrm{x}_{A_3})^2 + (y_k - \mathrm{y}_{A_3})^2} \\ \sqrt{(x_k - \mathrm{x}_{A_4})^2 + (y_k - \mathrm{y}_{A_4})^2} \\ \sqrt{(x_k - \mathrm{x}_{A_5})^2 + (y_k - \mathrm{y}_{A_5})^2} \\ \sqrt{(x_k - \mathrm{x}_{A_6})^2 + (y_k - \mathrm{y}_{A_6})^2} \end{bmatrix} \tag{4.35}$$

Where;

- x_k - Cartesian co-ordinate x at time k (m)

- y_k - Cartesian co-ordinate y at time k (m)
- x_{A_i} - Cartesian co-ordinate x of static Anchor $i \in [1 \ \ldots \ 6]$ (m)
- x_{A_i} - Cartesian co-ordinate y of static Anchor $i \in [1 \ \ldots \ 6]$ (m)

The observation covariance matrix is then constructed using the error metrics generated in Section 4.4.2.2 for the dynamic testing of the UWB system. It is to be noted that this observation covariance matrix is an additive noise value, meaning that it remains constant and is not adaptive in relation to the state vector components. The observation covariance matrix is shown in Equation (4.36), where $\sigma_{r_{Ai}}$ denotes the standard deviation of error for Anchor i.

$$R = \begin{bmatrix} \sigma^2_{r_{A1}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{r_{A2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{r_{A3}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{r_{A4}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{r_{A5}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{r_{A6}} \end{bmatrix} \tag{4.36}$$

Similar to the process conducted for the motion model, due to the definition of the system being of a non-linear nature, there is an equal requirement for the linearisation approximation of the observation model. This is also achieved through the first-order Taylor expansion represented as a Jacobian matrix of the observation function h with respect to the state vector χ. The standardised form of the observation model Jacobian H_{z_k} is shown in Equation (4.37).

$$H_{z_k} = \frac{\partial h}{\partial \chi}(\widehat{\chi}_k^-) \tag{4.37}$$

It is also evident that the observation function is also of a non-linear nature, where the change in the output predicted range estimation is disproportional to the change in robot position. This also clarifies the necessity of the EKF for the state estimation of a non-linear system. The formulation for H_{z_k} is shown in Equation (4.38).

$$H_{z_k} = \begin{bmatrix} \frac{x_k - \mathrm{x}_{b1}}{\sqrt{(x_k - \mathrm{x}_{b1})^2 + (y_k - \mathrm{yb}_1)^2}} & \frac{y_k - \mathrm{y}_{b1}}{\sqrt{(x_k - \mathrm{x}_{b1})^2 + (y_k - \mathrm{y}_{b1})^2}} & 0 \\ \frac{x_k - \mathrm{x}_{b2}}{\sqrt{(x_k - \mathrm{x}_{b2})^2 + (y_k - \mathrm{y}_{b2})^2}} & \frac{y_k - \mathrm{y}_{b2}}{\sqrt{(x_k - \mathrm{x}_{b2})^2 + (y_k - \mathrm{y}_{b2})^2}} & 0 \\ \frac{x_k - \mathrm{x}_{b3}}{\sqrt{(x_k - \mathrm{x}_{b3})^2 + (y_k - \mathrm{y}_{b3})^2}} & \frac{y_k - \mathrm{y}_{b3}}{\sqrt{(x_k - \mathrm{x}_{b3})^2 + (y_k - \mathrm{y}_{b3})^2}} & 0 \\ \frac{x_k - \mathrm{x}_{b4}}{\sqrt{(x_k - \mathrm{x}_{b4})^2 + (y_k - \mathrm{y}_{b4})^2}} & \frac{y_k - \mathrm{y}_{b4}}{\sqrt{(x_k - \mathrm{x}_{b4})^2 + (y_k - \mathrm{y}_{b4})^2}} & 0 \\ \frac{x_k - \mathrm{x}_{b5}}{\sqrt{(x_k - \mathrm{x}_{b5})^2 + (y_k - \mathrm{y}_{b5})^2}} & \frac{y_k - \mathrm{y}_{b5}}{\sqrt{(x_k - \mathrm{x}_{b5})^2 + (y_k - \mathrm{y}_{b5})^2}} & 0 \\ \frac{x_k - \mathrm{x}_{b6}}{\sqrt{(x_k - \mathrm{x}_{b6})^2 + (y_k - \mathrm{y}_{b6})^2}} & \frac{y_k - \mathrm{y}_{b6}}{\sqrt{(x_k - \mathrm{x}_{b6})^2 + (y_k - \mathrm{y}_{b6})^2}} & 0 \end{bmatrix} \tag{4.38}$$

The formulation presented in Equation (4.38) shows that the third column is a vector of zeroes. The reason for this is due to the 2D pose heading component ψ not being updated by the external UWB observations and is only calculated within the motion model. It is understood that it is possible to estimate a change in heading based on the corresponding change of Cartesian position. However, due to the occasional volatility of the UWB observations, a decision was undertaken to calculate the heading updates using the gyroscopic outputs only to restrict the possibly of fluctuating heading calculations.

4.5.4 Outlier Detection & Mitigation

As previously presented in Section 4.4.2 where the variance of the UWB system's range observations are assessed, these observations are occasionally prone to outliers which can have an overall effect on the estimated states of the unmanned system. When assessing the UWB range measurements in Section 4.4.2, a general population was collected and analysed post process. This enabled the acquisition of a mean and standard deviation for the calculation of the Z score metric, which beneficially enabled the identification and removal of the outlier data points. However, as the EKF is an algorithm that is intended to be deployed in real time, it is not possible to acquire a population mean and standard deviation. Therefore, an outlier detection process is required to operate upon each observation acquisition iteration. This process enables outliers to be detected and dealt with correctly. This outlier detection and mitigation process is separated into two

steps presented in Section 4.5.4.1 and Section 4.5.4.2.

4.5.4.1 Identification

The outlier identification technique deployed is known as the Mahalanobis Distance (MD) metric. MD is a generalised measure of how many standard deviations a point is away from the mean of a distribution by calculating a normalised distance to the mean [58]. MD differs from the utilisation of the Euclidean Distance (ED) by not only taking the distance but also its covariance. This technique has been deployed in certain robotics based applications such as outlier and anomaly detection [37] and landmark matching for robust SLAM deployments [59]. The general form for the calculation of the MD is shown in Equation (4.39)

$$MD(x) = \sqrt{(x_i - \bar{x})^T C^{-1} (x_i - \bar{x})} \tag{4.39}$$

The deployment of MD for this specific application follows that included in [37], where the predicted set of measurements $\widehat{Z}_{Bi}$ is assumed to be the statistical population and is therefore deployed as the mean ($\bar{x}$) in Equation (4.39). As this stage of the R-EKF is to identify outlier measurement from the UWB system, the observations Z_{Bi} are used as the input into Equation (4.39) and the covariance value implemented is S_k, which is defined as the residual covariance that represents the combined uncertainty of the predicted range measurements and the actual range measurements acquired from the UWB anchors. The final implementation of MD for range-based outlier detection is presented in Equation (4.40).

$$MD(Z_{Bi}) = \sqrt{(Z_{Bi} - \widehat{Z}_{Bi})^T S^{-1} (Z_{Bi} - \widehat{Z}_{Bi})} \tag{4.40}$$

If the calculation of MD exceeded a tuned threshold, then it was deemed as an outlier and then was passed to an imputation stage of the algorithm. This is explained in the following subsection.

4.5.4.2 Imputation

Applying a simple removal of data points classified as outliers will effectively have lasting reduction effects on parameters such as the overall RMSE, however, it will consequently reduce the total sample size of the resulting EKF output. This reduction in the dataset can

consequentially result in the statistical inaccuracy of the comparison between the R-EKF and all other possible indoor localisation solutions. Therefore, if an observation iteration was deemed as an outlier, the imputation technique deployed was the execution of the prediction stage of the R-EKF. Deploying this method therefore prevents the necessity to totally disregard this sensor observation and instead to predict the next state usually the control input and the kinematic model. As a consequence, the covariance within the state belief would increase, however, this would be corrected once a new and reliable observation is received.

4.5.5 R-EKF Final Overview

Prior to the R-EKF implementation, an overview of the process of the algorithm is shown in Algorithm 3. The components of the R-EKF that are not defined in the above formulation are also noted. These variables have been presented in the Literature Chapters of this thesis concerning Gaussian based filters, however, they are reproduced here for reference simplicity.

Where;

- $\widehat{P}_k^-$ - Priori State Covariance
- $\widehat{P}_k$ - Posteriori State Covariance
- K - Kalman Gain
- Y - Observation Residual
- S_k - Residual Covariance
- $\widehat{Z}_{Bi}$ - Observations

4.6 Conclusion

To conclude, Chapter 4 as a whole presented the assessment of the 2D localisation problem for an unmanned ground vehicle. It demonstrated how the combination of a sound sensor measurement error characterisation process and effective state-space mathematical modelling can aid in the design of a peer reviewed state estimation system for unmanned robotic systems. The most beneficial aspect and result concerning the dynamic and remote

Algorithm 3 Range based EKF Localisation

Prediction:

$\widehat{\chi}_k^- = f(\widehat{\chi}_{k-1}, u_k)$

$G_{x_k} = \frac{\partial f}{\partial x}(\widehat{\chi}_k^-, u_k)$

$G_{u_k} = \frac{\partial f}{\partial u}(\widehat{\chi}_k^-, u_k)$

$Q_{u_k} = G_{u_k}\ U\ G_{u_k}^T$

$\widehat{P}_k^- = G_{x_k}\widehat{P}_{k-1}G_{x_k}^T + G_{u_k}QG_{u_k}^T$

Correction:

$\widehat{Z}_{Bi} = h(\widehat{\chi}_k^-, A_{B_i})$

$H_{z_k} = \frac{\partial h}{\partial x}(\widehat{\chi}_k^-)$

$S_k = H_{z_k}\widehat{P}_k^- H_{z_k}^T + R$

if *measurement_is_outlier* **then**

    do **Prediction**

else

    $K_k = \widehat{P}_k^- H_{z_k} S^{-1}$

    $Y_k = Z_{Bi} - \widehat{Z}_{Bi}$

    $\widehat{\chi}_k = \widehat{\chi}_k^- + K_k Y_k$

    $\widehat{P}_k = (I - K_k H_{z_k})\widehat{P}_k^-$

end if

if ! *measurement_is_available* **then**

    do **Prediction**

else

    do **Correction**

end if

ground truth acquisition technique is its ability to provide a reliable reference dataset for the unmanned system as it operates in its intended environments, therefore enabling the opportunity to thoroughly experiment with sensors in conditions that they would be expected to encounter, further improving the validation of conducted experimental methods. However, even though the unmanned system is ground-based it is expected to encounter scenarios where it will experience dynamic changes in motion in 3D not just 2D. Modelling

the unmanned system in 3D will enable the system to be robust in its deployment as it is prepared to adapt and to model motion of any variant. Therefore, the following Chapter 5 builds upon the work presented in Chapter 4 to model the system in 6DOF, which includes the six components of 3D motion both linear and angular displacements. It also utilises the RTS as a ground truth and finalised algorithm validation technique.

Chapter 5

Six Degree-of-Freedom Ego-Motion Estimation

Six degree-of-freedom (6DOF) motion is defined as the change in linear and angular state of a system in a 3D co-ordinate frame of reference. When considering point cloud acquisitions from LiDAR systems, the importance of 6DOF motion estimation is related to the transformation between each acquired point cloud to complete the registration and map building process. Scaled point clouds are modelled as rigid body structures, therefore they preserve their shape and size and any transformations of rigid bodies include alternations to translational and rotational states in 3D and are applied through a rotational transformation matrix. The rigid body transformation matrix is represented by a [3 x 3] rotation matrix parameterised by the 3 Euler Angles $[\theta\ \phi\ \psi]$ that act on each axis in a 3D reference frame and a [3 x 1] translational column vector $[x\ y\ z]$. Therefore, the combined vector $[x\ y\ z\ \theta\ \phi\ \psi]^{\boldsymbol{T}}$ denotes the components of 6DOF motion.

Following on from the 2D Localisation problem analysed in Chapter 4, the estimation of motion in 2D provided an initial insight into how the consideration of a motion model, the error handling of model inaccuracies and measurement inputs, can be utilised in the design of an effective state estimation system. However, this only assesses half of the problem as achieving the design of a robust autonomous registration and map generating process using mobile LiDAR systems requires the estimation of 6DOF motion between two sequential point cloud frames $\boldsymbol{P}_k$ and $\boldsymbol{Q}_k$.

In this section, a two-step method is assessed which estimates the motion of the LiDAR system and the mobile robot using the feedback acquired from a LiDAR itself. As discussed, the importance of 6DOF motion estimation is an influential factor to consider for the

registration procedure. The two-step method presented in this chapter utilises the rapidly scanning lasers for odometry and 6DOF motion estimation. The first of the two steps executes dominant ground plane alignment process where the ground planes in sequential scans are identified, isolated and assessed to provide updates concerning linear displacement on the z axis and angular displacement for orientations concerning the system's roll and pitch Euler angle representations, this is noted in column vector form as $[\theta\ \phi\ z]^T$. The second step operates a 2D feature matching process to provide updates concerning linear displacements in x and y and angular displacements for the system's heading, represented in vector form as components $[x\ y\ \psi]^T$. Chapter 5 provides information concerning the issues faced and how they were overcome through the introduction of an Error State Kalman Filter (ES-KF) which was deployed as a sub-system within the 6DOF-EKF.

5.1 State of the Art

Section 5.1 presents the brief state of the art concerning 6DOF motion estimation in general and then expands upon this topic into a more focused area of utilising LiDAR for 6DOF motion estimation. Section 5.1 then seeks to present the state of the art concerning the generalised combination 6DOF motion estimation and the use of an RTS for active tracking and motion estimation validation. Finally, it then specifically targets state of the art concerning the integration of LiDAR for 6DOF motion estimation, all whilst using an RTS for output validation.

5.1.1 6DOF-EKF Pose Estimation

6DOF pose estimation is a key topic within this chapter, therefore, Section 5.1.1 presents the state of the art for this topic. It begins with the consideration of the robotic systems developed by Boston Dynamics [60]. Boston Dynamics are the forefront of sophisticated autonomous robot deployment, whether that is robots that are legged, wheeled or a combination of both, through the ground breaking control systems and state estimation algorithm design. They are especially known for the design of an autonomous and intelligent humanoid [61]. An initial study conducted by a collaboration between Boston Dynamics and California Institute of Technology (CALTECH) presented the design concerning a robust multi-sensor 6DOF pose estimation system for legged vehicles in GNSS-denied environments

[62]. Within the design of this control system, the sensor fusion process consisted of leg odometry to estimate the current state of each leg, an IMU for robot head position and a secondary IMU for estimation of states relating to the robot's body. These proprioceptive inputs were then fused with a stereo camera deploying a VO technique. The authors decided towards a probabilistic approach for the fusion of these multiple sources of sensory data through an EKF, with the inclusion of a kinematic motion model and a 15 element state vector. The integration of an EKF for the design of a sophisticated and non-linear system such as the legged robot included within this study demonstrates the power and applicability that the EKF algorithm possesses. The results gathered from this study yield that the designed EKF achieves a 6DOF pose estimation of less than 0.5m for every 50m travelled. This study is perceived as the foundational aspects of the Boston Dynamics system known as the SpotMini [63].

Moving on from the one of the main commercial implementations towards more of a research and development focus, a popular method to provide 6DOF motion estimation is through the use of visual-based sensors which are recursively perceiving the dynamic feature changes in the environment. As briefly mentioned in the discussion of the Boston Dynamics system, this vision-based technique is approached through the deployment of a monocular visual odometry system (MVO), where a single camera is utilised, or a stereo visual odometry (SVO) which uses two cameras that are separated by what is known as a baseline. The underlying difference between MVO and SVO techniques is the inclusion of such a baseline. The baseline is defined as the distance between the two cameras within a SVO set-up, therefore, providing the system with the ability to scale the environment, a characteristic that MVO system's alone do not possess without the inclusion of extra depth perception sensors. A study which is a good example of the integration of VO for 6DOF motion estimation is one conducted by *Zhu et, al* [64], where event based visual odometry and an IMU are utilised within an EKF for 6DOF camera motion estimation. Visual odometry seeks to estimate the rotation and translation between each captured image frame based on features within the environment. The major findings within this study indicate that the event based visual odometry algorithm probabilistically fused with IMU output is capable of estimating long camera trajectories in 6DOF with minimal drift.

A similar study concerning the use of VO for pose estimation is one conducted by *Alatise et.al* [65], who presented the 6DOF pose estimation of a mobile robot using vision

perception fused with an IMU through the utilisation of an EKF. Within this study, *Alatise et.al* deployed a technique known as speeded-up robust features (SURF) [66], which is a monocular vision-based object and feature detection algorithm. *Alatise et.al* state that the fusion of observations acquired from inertial and vision-based systems provide a more robust pose tracking estimation and overcome inadequacies associated with individual component-based tracking. An example of this is scenarios where the mobile robot experiences fast movements, where the camera will be subjected to motion blur. The inertial nature of an IMU can aid in the reduction of this occurrence when fused with the vision system through some state estimation algorithm such as the EKF. The major findings were that a hybrid approach of the fusion between inertial sensors and vision yields a much improved 6DOF pose estimation in comparison to other traditional vision-based techniques.

The aforementioned studies have presented the integration of vision-based systems fused with other sensing units through EKF implementations. However, more studies have occurred which have employed alternative methods. An example of this is the study conducted by *Hol et.al* [67]. *Hol et.al* deploy an EKF fusion of UWB for positioning and IMU for orientation. As presented in Chapter 4, UWB functions through a network of nodes each providing range measurements between reference nodes and a dynamic rover node and an algorithm of a tri-lateration nature is then used to calculate position. However, this does not provide information concerning 3D orientation of the platform which it mounted to, hence why *Hol et.al* combine UWB information with an IMU. This is similar to the work conducted in Chapter 4 of this thesis, however, it was conducted in a 2D sense as opposed to 3D. *Hol et.al* state that the results acquired from the 6DOF pose estimation and tracking through UWB/IMU sensor fusion achieve a robust estimation result and adapt to scenarios where CLOS conditions for the UWB system are not present. The UWB system alone cannot provide estimations for 6DOF pose estimation without the inclusion of the IMU, something which vision-based systems can achieve initially without an IMU present.

A study conducted by *Simanek et.al* [68] offers a different approach to the state of the art studies already presented in this section. Instead of designing the EKF around a certain observation source and robotic system, it takes a pre-built system and sensor input and evaluates different state estimation architectures that focus on EKF deployment. *Simanek et.al* compare the output for the non-linear model (NLM) and the error model (ERM) and both architectures combined with a complementary filter (CF), which effectively

operates as a moving average. The results gathered from this study state that the use of the NLM method outperformed the ERM for all 6DOF components of motion, where the NLM approach is used in the design of the 6DOF-EKF in this chapter.

Due to the limitations of UWB and its requirement to use an IMU to provide 6DOF pose estimation capabilities, and vision-based systems' limitations in environments that possess no light or low lighting conditions, a decision was undertaken to deploy a LiDAR system as the main observation source to estimate 6DOF pose of a mobile robot. This also aids in the development of positioning systems suitable as alternative solutions to GNSS in indoor environments. Implementations of EKF and LiDAR are outlined in the following section.

5.1.2 6DOF-EKF Ego-Motion Estimation with LiDAR

A well known study was conducted by *Zhang et.al* who developed an odometry estimation and mapping algorithm known as LOAM, which is short for Laser Odometry And Mapping [69]. In the initial version of the study, *Zhang et.al* utilised a 2D sweeping LiDAR which was set to operate with 6DOF motion. The designed LOAM algorithm estimates 6DOF motion through the assessment and registration of recursively produced point clouds acquired from the LiDAR. This registration process is achieved by identifying corresponding features between two consecutive point cloud scans generated from the LiDAR such as edge point detection and surface plane patches. This feature corresponding technique is conducted on consecutive point cloud scans to output a 6DOF transformation between the two scans, therefore outputting an estimation of 6DOF per acquired scan. The findings from this study state that the algorithm produced a lower level of scan matching error when operated in indoor environment such as lobby and corridor areas as there was more access to distinctive feature points. Error production was calculated at approximately 1.1% of a total distance of 58m and 2.8% of a total distance of 67m for indoor and outdoor environments respectively. However, *Zhang et.al* progressed their algorithm to incorporate an IMU and a vision-based odometry stage to improve overall motion estimation [70]. This variant of the algorithm was presented as Visual-LiDAR Odometry And Mapping (V-LOAM). The V-LOAM algorithm functions through two threads operating in conjunction and fusing the output from both for correction and error reduction. The initial stage consists of a frame to frame motion estimation at 60Hz update rate using collected images within the visual odometry stage

of the algorithm. These estimations are then utilised as input into the LiDAR odometry state for sensor fusion and drift error correction. The data acquired from the addition of the visual odometry stage yielded lower errors in terms of motion estimation and drift, however, *Zhang et.al* states that in environments with low-light conditions for considerable amounts of time, the V-LOAM method is unsuitable and experiences limitations. Nonetheless, the V-LOAM algorithm has now been commercialised as an integrated mapping system available as a GNSS-less mapping solution [71].

A study conducted by *Pandey et.al* [72] presents a technique designed with the aim to align 3D point clouds that possess a dominant ground plane. This algorithm would be suitable for ground-based systems such as UGVs and autonomous consumer vehicles. In this study, *Pandey et.al* state that to align 3D point clouds is to extract 6DOF transformation between two consecutive point clouds, and they utilise a ground plane alignment application in conjunction with a mutual information (MI) approach to achieve this. MI between is a technique which analyses two random variables either acquired from a single or multiple sensory sources and calculates the measure of the statistical dependence occurring between them. This is commonly achieved through the entropy of the random variables in question. In the case of this study, *Pandey et.al* utilise five sources of data acquired from a LiDAR and an omnidirectional camera unit mounted to a commercial automobile. These five sources of information are reflectivity, Z-height and point normals acquired from the LiDAR and the RBG and grey scale information acquired from the vision unit. The findings state that the algorithm yields sub 25cm translational and rotation error until there is a 7m and 15° difference between two consecutive scans respectively and outperforms traditional registration techniques. It is understood that this study is approached with the aim to scan an environment and does not currently seek to integrate such probabilistic robotic state estimation approaches.

SLAM is the problem of deploying an unmanned system to dynamically construct a map of the environment whilst possessing the capability to simultaneously localise itself within that map [73]. Many approaches undertaken suggest that the unmanned system is modelled as a rigid body, therefore it is affected by 6DOF motions as it navigates. Therefore, many 6DOF SLAM applications have been conducted which utilise a Bayesian inference approach through the deployment of Gaussian filters for probabilistic state estimation, such examples are known as EKF-SLAM. A study conducted which falls into this category is one conducted

by *Cole et.al* [74], which focuses on the use of LiDAR for an EKF based SLAM application for exploration of outdoor environments.

A similar study concerning the integration of EKF and SLAM is one conducted by *Weingarten et.al* [75]. *Weingarten et.al* formulate a motion model based on a differential drive unmanned system using a 2D laser scanner as a an observation source through segmented feature extraction and planar surface extraction. The state vector within this study is set to contain all 6DOF components of the robot. The major findings achieved demonstrate how an EKF-based SLAM algorithm can be used in 3D space for 6DOF motion estimation. It also demonstrates how the combination of kinematic motion modelling and 3D assessment of the environment can significantly reduce drift error produced from standard robot odometry.

Although the aforementioned examples are of research applications concerning 6DOF motion estimation through implementations of the EKF, SLAM and unmanned robotics, a unique study conducted by *Bosse et.al* [76] offers an alternative approach. *Bosse et.al* focus on the deployment of a spring mounted 3D LiDAR system which can either be hand-held, situated within a backpack or mounted to head wear for mobile mapping applications. This study focuses on a SLAM execution for 6DOF motion estimation of the LiDAR whilst mounted to the head of the spring. This study is unique as the LiDAR is not rigidly mounted to the surface of an unmanned platform and therefore does not rely on the motion of the platform it is mounted to. In fact, the LiDAR relies on the bouncing dynamics of the spring and a SLAM approach is employed to estimate motion from thereon. In order to conduct a quantitative analysis, *Bosse et.al* utilised a Vicon motion capture system, which consists of a network of 14 cameras that can track an object with millimetre precision to provide a 6DOF ground truth metric. Major findings state that the custom SLAM algorithm produces trajectories with sub-centimetre accuracy in comparison to the Vicon visual tracking system. However, *Bosse et.al* state that there are also limitations to the system in terms of traditional issues with SLAM applications, such as open plan environments and featureless corridors. Additionally, the developed SLAM algorithm is tasked only to track the head of the bouncing spring, if this system was to be mounted on unmanned systems, issues arise in relation to estimating the position of the spring head in relation to the current position estimation of the unmanned system that it is mounted to. In short terms, this is the problem of robotic co-ordinate reference frame transformations.

This algorithm has now also been developed into an integrated mapping system and is commercially accessible [77].

The items of literature that have been presented in this section provide state of the art studies concerning LiDAR and its integration with other sensory systems for the task of 6DOF pose estimation, with its application to robotic and non-robotic systems.

5.1.3 6DOF-EKF Pose Estimation Validation

Within this thesis, the RTS is used as a means to optically track unmanned systems for validation of the designed state estimation algorithms. Therefore, Section 5.1.3 outlines the state of the art in relation to this topic, both with or without the integration of robotic state estimators such as the EKF.

A popular method to provide ground truth metrics and validation is through the use of a Vicon system, which was briefly mentioned in Section 5.1.2. The Vicon system is a motion capture network of cameras which tracks reflective targets or objects which are mounted to the platform that is required to be tracked [78]. A study conducted by *Merriaux et.al* focused on the positioning performance of a Vicon system [79]. Although the Vicon system is considered as the market leader in object tracking in relation to performance, [79] deployed a 4D numeric controlled robot which can attain 10μm positional resolution to enable a quantitative analysis of the Vicon's performance. Although this numerically controlled robot is ideal for this testing scenario, however, its deployment for field robotics applications is unsuitable due to its static nature and its dynamic inabilities. However, it does possess the ability to simulate unmanned system movement on a smaller scale. The findings acquired from *Merriaux et.al* state that the Vicon achieves a mean position error of 0.15mm with a 0.015mm standard deviation statically, however, it achieves 2mm when operating dynamically. Therefore, the update rate of the Vicon system needs to be carefully chosen in relation to object operating speeds. Nonetheless, Vicon systems have been utilised for various non-robotics based studies such as applications concerning its use within sporting activities [80], GNSS reliability [81] and validation for human body orientation estimation through the implementation of an EKF [82].

In relation to robotics and the topic of 6DOF pose estimation, a study conducted by *Hol et.al* which was also covered in Section 5.1.1 is an example of how a Vicon system can be deployed for validation. This study, however, utilises high performance and high cost

IMUs and does not obtain a full controlled loop system.

Chen et.al [83] conducted a study that focused on EKF based localisation for ground operating mobile robotic systems. This study consisted of a traditional odometry motion, identical to the motion model deployed in this thesis. The odometry motion model sets lateral constraints to the system and therefore is able to filter out ambiguous lateral motion. Within the design of the EKF state estimator, a laser range finder is used as the main observation source of the mobile robot. In order to acquire information concerning the temporal transitions of motion, *Chen et.al* use a line fitting approach to identify corners and planar surfaces perceived from the environment and then fuse this with odometry within the EKF. The output acquired from the EKF was then validated through a quantitative analysis process using the Vicon as the ground truth metric. A study conducted by *Shen et.al* [84] deploys a Vicon system for motion tracking of a micro aerial vehicle (MAV) for autonomous multi-floor indoor mapping. *Shen et.al* state that the design of the 6DOF pose estimation algorithm is conducted through an EKF-SLAM based approach, where a sensor fusion process is applied combining a LiDAR, camera and IMU all whilst identifying a loop closure for trajectory and attitude optimisation. In the case of this study, *Shen et.al* state that the Vicon was taken as the ground truth due to its sub-millimetre accuracy, therefore demonstrating the capabilities of the Vicon for 6DOF pose estimation of non-linear unmanned systems both aerially and ground operating.

A study which relates to the development of a 6DOF-EKF motion estimation system using a RTS is one conducted by *Keller et.al* [85]. *Keller et.al* seek to design an EKF tasked with motion estimation for indoor mapping as the acquisition of high precision and standard GNSS services are inaccessible in indoor environments. Although this study does not deploy an RTS for EKF validation, however, it proposes the RTS as an observation source for the EKF. The definition of the measurement covariance for the RTS, which is considered as an influential value in the design of an EKF, is uncommonly, not stated. The use of the RTS for an observation source differs from the work conducted in Chapter 5 of this thesis, as *Keller et.al* did not use the point cloud data acquired from the on-board LiDAR system as a means to provide motion estimation, the point cloud data is solely used as a means to generate 3D maps of the environment. Therefore, in order for their system to function, an RTS is required to be configured upon each and every indoor mapping exploration. This may arise as a limitation within this study as with the RTS being a statically based

instrument, unpredictable indoor environments such as acute angles, physical barriers and turning corners may result in the loss of RTS active tracking.

5.2 Light Detection and Ranging

Light Detection and Ranging, commonly referred to by the acronym LiDAR, is an optical measurement technique which incorporates the use of light in the form of rapidly pulsing lasers that can possess the ability to operate up to pico-second (ps) rate. A LiDAR system will execute a scanning process which consists of lasers firing in sequences with an aim to measure varying distances to objects within the surrounding environment. The fundamental operation of a LiDAR system that enables it to measure distances using laser light is based on a pulse-echo operation. Lasers originating from the LiDAR device are pulsed with an aim to reflect from objects within the environment, once a laser is pulsed, an internal clock sets a timer to operate until that laser is echoed off an object and returns. The time taken for the laser to return is related to the finite speed of light to determine a distance to that certain object. This is known as the TOF and is shown in Equation (4.1). Typical LiDAR devices are comprised of the main components that are each outlined in the list below;

- Laser Source, Detectors & Scanning Mechanism

  The laser source generates the energy which pulses the lasers into the environment. Most commercial LiDAR systems operate on near infra-red wavelengths and are categorised as Class 1 Eye-Safe due to the nature of their deployment [86], however, this is not always the case as bathymetry LiDAR systems can deploy lasers within the blue green section of the electromagnetic spectrum, thus enabling sea water penetration to map the sea bed [87]. The method of registering distances to objects using the lasers solely relies on the capabilities of the detector components of the device. The rapidly pulsing lasers deployed can operate to acquire one million distance measurement readings per second [88]. In order to achieve this data acquisition frequency and low erroneous distance measurements, the internal components functioning to identify echoed laser returns have to be sophisticated enough to operate at such high speeds. As lasers are technically defined as intense beams of light stimulated by an emission of photons, avalanche photodiodes (APD) or Silicon APDs are traditionally employed to detect pulsed lasers and echoed returns

[88]. APDs are semiconductor electronic devices which convert detected light photons into an electric current, where the generated current is used to signal the timing electronics to log the departing time of a pulsed laser and the initial reception time of an echoed return. The performance of the APD therefore has a great effect on the accuracy of the distance measurement as timing parameters are of the greatest importance [88].

- Internal Micro Controller Unit & Timing Electronics

  The internal MCU within the LiDAR device is tasked with triggering the laser driver to deploy a pulsed laser, signalling the beginning of a firing cycle. Prior to the laser being pulsed, the MCU also sends a start signal to a component called Time-Digital Converters (TDC). A TDC is a device which measures the time interval between a start and stop pulse and converts the output to a digital (binary) format. Once the laser is reflected off an object and is detected by the photo diode, the current produced from the photodiode is then passed into a transimpedance amplifier (TIA), which converts current into a voltage. This voltage is then used to signal the TDC indicating the acquisition of an echoed laser. The interval between the start and stop pulses is then used to calculate the distance to the object.

- Positioning System

  The LiDAR device is traditionally operated with a high precision differential GNSS or Real Time Kinematic (RTK) GNSS for geo-referencing the data acquired from the LiDAR system. Both the LiDAR point cloud and the output from the positioning system are joined together and referenced to a local, fixed-earth coordinate system. This enables collaborative data collection between multiple LiDAR devices as the points collected all are converted within the same reference co-ordinate system, as well as identifying areas of the environment which have been previously scanned [89]. However, as GNSS is known to possess limitations in RF perturbed environments such as mountainous terrains and indoor scenarios [90, 91] many alternative methods of tracking the position of the LiDAR device have been implemented.

A graphical representation of the firing cycle for the laser and how the components mentioned above are all operating in conjunction is shown in Figure 5.1.

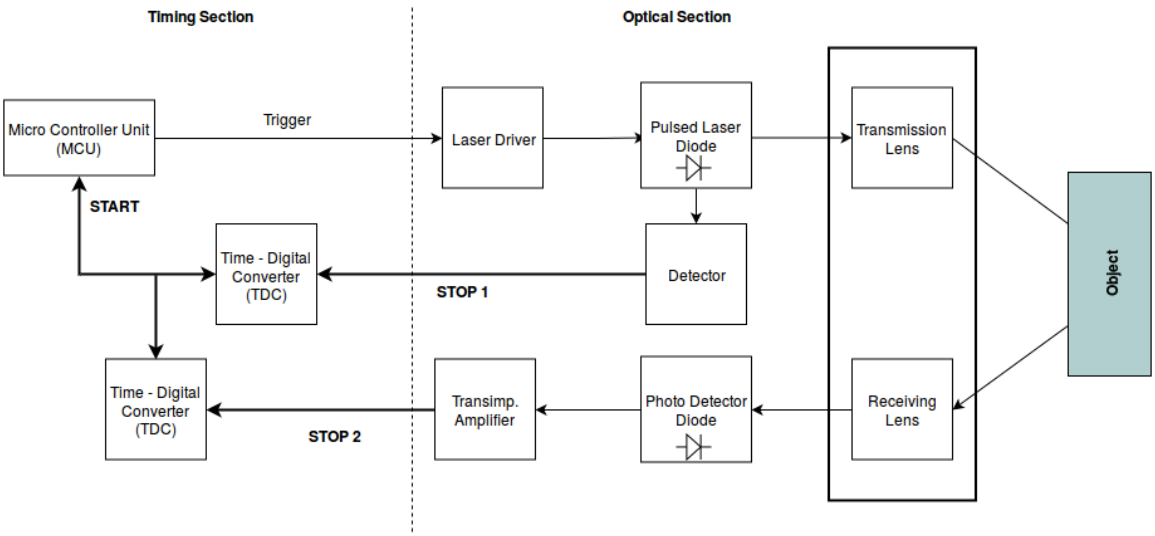


Figure 5.1: LiDAR device operating procedures

The following sections provide discussions on mapping with LiDAR, positioning of the device for map generation and different types of LIDAR systems that are commonly used. This is the main section as to how a LiDAR device is utilised for the work conducted in Chapter 5.

5.2.1 LiDAR Mapping

The sequential firing of the lasers to acquire raw range measurements to objects is an important factor in the area of mobile robotics. The ability to produce maps of the environment that a mobile robot is situated within enables the design of autonomous navigating systems. For the case of the work conducted in this thesis, a Veldoyne VLP-16 LiDAR system was utilised [92]. The VLP-16 is a lightweight, compact and robust LiDAR system that is mountable to any unmanned system whether aerial or ground-based. The VLP-16 deploys 16 fixed firing lasers providing 30° vertical field of view (FOV) and it functions through an internal rotating column, therefore providing a full 360° horizontal FOV. When combining the range measurements acquired from the VLP-16 with both the vertical and horizontal firing angles of each laser, the calculation of a point with its own 3D co-ordinate system is achievable. This process consists of a Polar to Cartesian conversion which is shown in Figure 5.2, along with the respective equations for calculating each component of the 3D point shown in Equations (5.1) to (5.3).

$$X = R\ cos(\omega)\ \ sin(\alpha) \tag{5.1}$$

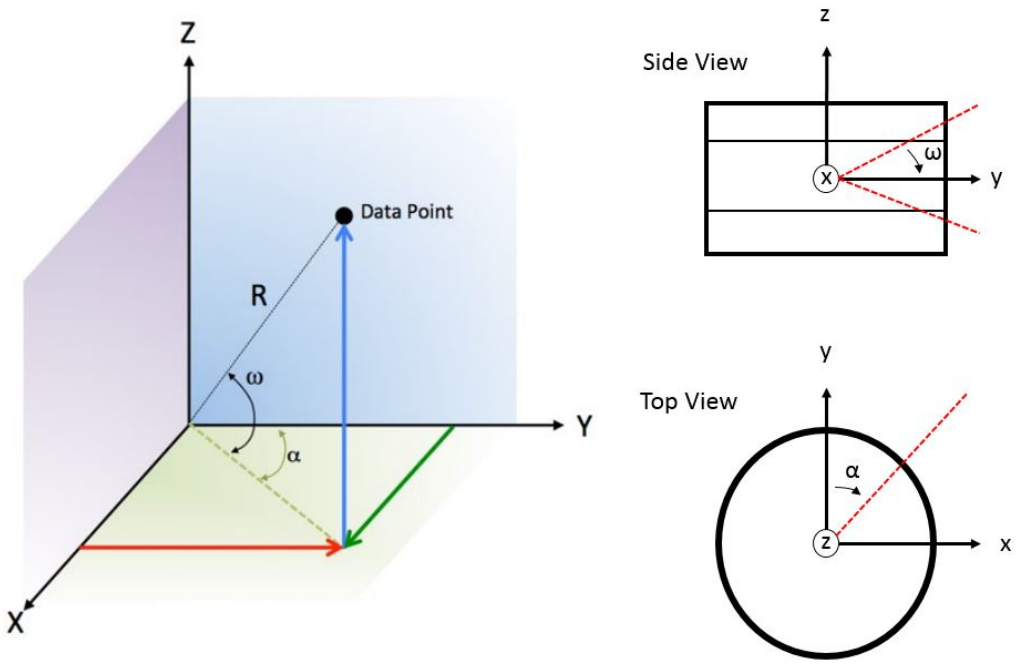


Figure 5.2: Polar - Cartesian Conversion [92]

$$Y = R\ cos(\omega)\ cos(\alpha) \tag{5.2}$$

$$Z = R\ sin(\omega) \tag{5.3}$$

Where;

- R - Range measurement (m)
- ω - Vertical firing angle (deg)
- α - Horizontal firing angle (deg)

The rapid 3D point collection of a LiDAR system results in the generation of a point cloud, where a point cloud can be defined as a cluster of points each with their own 3D spatial co-ordinate system. A point cloud is the typical format outputted from a LiDAR device and outlined below is a list of the key components that are either a characteristic of a point cloud or an influential factor in their generation.

- Range (m)
- Elevation (Vertical Angle to Point ω) (deg)
- Azimuth (Horizontal Angle to Point α) (deg)

- Spatial 3D Co-ordinates (X,Y,Z) (m)
- Intensity of return

5.2.2 Types of LiDAR

The deployment of different types of LiDAR systems for mapping is dependent upon factors such as the exploration plan, intended operating environments and assessment concerning the ideal outcome of the LiDAR survey. For example, an exploration plan consisting of mapping a large urban area would require a mobile terrestrial or possibly an airborne LiDAR system to acquire data and produce maps at a more rapid rate. An overview of the three main types of LiDAR systems and deployment techniques are discussed below.

5.2.2.1 Static Terrestrial LiDAR

A static LiDAR system commonly known as a Terrestrial Laser Scanner (TLS) is a ground-based stationary LiDAR unit mounted upon a robust tripod. An example of a TLS is shown in Figure 5.3 which shows the commercially available FARO TLS [93]. Due to the static scanning nature of a TLS, the scanning mechanism deployed within is commonly through a rotating mirror which deflects laser beams to enable the TLS to scan on a vertical plane. The complete hemispheric rotation therefore enables the device to capture a 360° FOV. Due to the FARO TLS being designed to operate terrestrially and statically, the inclusion of such rotating mirrors is achievable as design considerations concerning payload weight and size is negligible. This differs from the mobile VLP-16 as it is designed to be deployed dynamically upon unmanned systems whether aerial, ground-based or even autonomous surface vehicles. Therefore, payload limitations need to be considered if the system is to be robust in its deployment. Some example applications of TLS implementations have been conducted in areas such as forestry mapping [94], vegetation analysis [95] and 3D urban environment modelling [96].

5.2.2.2 Mobile Terrestrial LiDAR

LiDAR systems that are designed to operate dynamically on ground-based vehicles are commonly referred to as mobile laser scanners (MLS), MLS is also used to define mobile laser scanning. MLS consists of the deployment of LIDAR systems to actively navigate an environment through perception or to generate 3D maps of the environment. The integrated

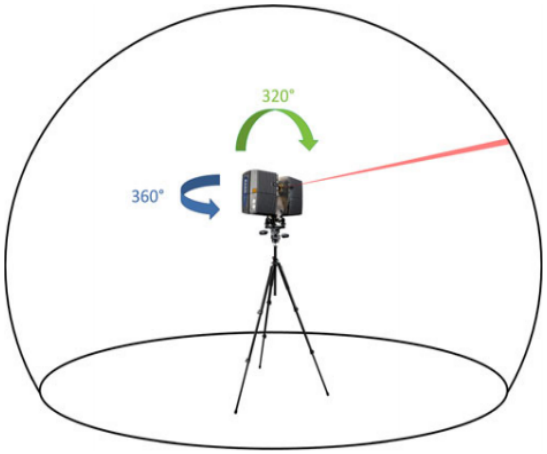


Figure 5.3: Static terrestrial LiDAR system [94]

mapping system shown in Figure 5.4 demonstrates a cluster of mobile LiDAR units, RGB cameras, high precision IMUs and a high precision RTK-GNSS unit for positioning and the possible registration of the point clouds acquired from the LiDAR [97]. This system is purely deployed for research purposes. It is to be noted that the design and deployment of such large integrated mapping systems as that shown in Figure 5.4 is possible if the automobile utilised for exploration is able to withstand heavy payloads. Therefore, proving the opportunity to mount a large variety of sensors for research based algorithm design and validation. MLS have been deployed for example in applications such as MLS data integration in neutral network design [98], MLS comparison with TLS for cultural heritage site dissemination [99] and MLS data segmentation and classification for features within urban environments [97].

5.2.2.3 Airborne LIDAR

With the recent advancements in commercial UAV technology providing more opportunities to explore large scale areas and the exploration of locations inaccessible to humans, remote sensory systems such as LiDAR devices have also adapted in terms of size, weight and cost effectiveness. Airborne LiDAR systems, commonly referred to as Airborne Laser Scanning (ALS), provides the opportunity to consumers to conduct mapping surveys with a greater height advantage. Figure 5.5 shows in image of a custom designed UAV with the VLP-16 LiDAR unit as its primary payload. The UAV is a sub-7kg system including all payload units and was equipped with a NVDIA miniature computer which operated

Figure 5.4: Mobile Terrestrial LIDAR system [97]

on a software level through a fully transparent data integration nodal network. This means that feedback observations from the on-board sensing systems which include the GNSS, IMU and LiDAR were all time-synchronised. Therefore, providing an example of a fully functioning airborne LiDAR data collection system. Previously conducted studies demonstrating possible applications for UAV based LiDAR systems include rainforest and landscape mapping [100], collaborative and integrated ALS and MLS mapping surveys [101] and review studies into the classification of urban land cover using ALS systems [102].

5.2.3 Conclusion

Section 5.2 presented an introduction to the theory of ToF based LiDAR and how it is deployed for 3D mapping applications. It also presented the three main techniques of LiDAR deployment and provided graphics which are examples for each deployment technique. The following section presented as Section 5.3 now progresses onto the demonstration of how 6DOF motion estimation is achieved using data provided from the VLP-16 LiDAR device.

Figure 5.5: Airborne integrated LIDAR system

Section 5.3 encompasses what will be the observation source into the 6DOF-EKF.

5.3 Two-step LiDAR Ego-motion Estimation

The main aim of Section 5.3 is to provide a demonstration of how the point cloud data acquisition is used as an observation source for the 6DOF-EKF. The raw point cloud data acquired from the LiDAR system undergoes transformations and matching procedures and this process is one of the vital stages of the overall 6DOF-EKF algorithm. The following sections outlined in Section 5.3.1 and Section 5.3.2 present each stage of the LiDAR odometry algorithm and an example of how the point clouds are processed in order to acquire estimations for each component of motion.

5.3.1 Dominant Ground Plane Extraction & Alignment

This section presents the second step of the LiDAR odometry algorithm. This step consists of the alignment of the dominant ground planes (X-Y plane) in two consecutive misaligned scans ($\boldsymbol{P}$, $\boldsymbol{Q}$). This step provides estimations on the roll, pitch and translational Z $[\theta\ \phi\ z]$ motion offset between $\boldsymbol{P}$ and $\boldsymbol{Q}$. Additionally, using the time update rate of the Velodyne LiDAR, it was also possible to provide estimations of the temporal transition of these components $[\dot{\theta}\ \dot{\phi}\ \dot{z}]$. The extraction of the ground plane is achieved using the Random Sample Consensus (RANSAC) plane identification algorithm. RANSAC is expanded upon in the following section [72].

To demonstrate ground plane isolation using RANSAC, a traditional point cloud acquired from the mobile Velodyne LiDAR system was manipulated to undergo simulated rotations around the X and Y axes as well as a translation in the Z. Such motions were applied to simulate the transition in the 3 components of 6DOF motion that this step of the algorithm estimates. The native Velodyne point cloud is shown in Figure 5.6.

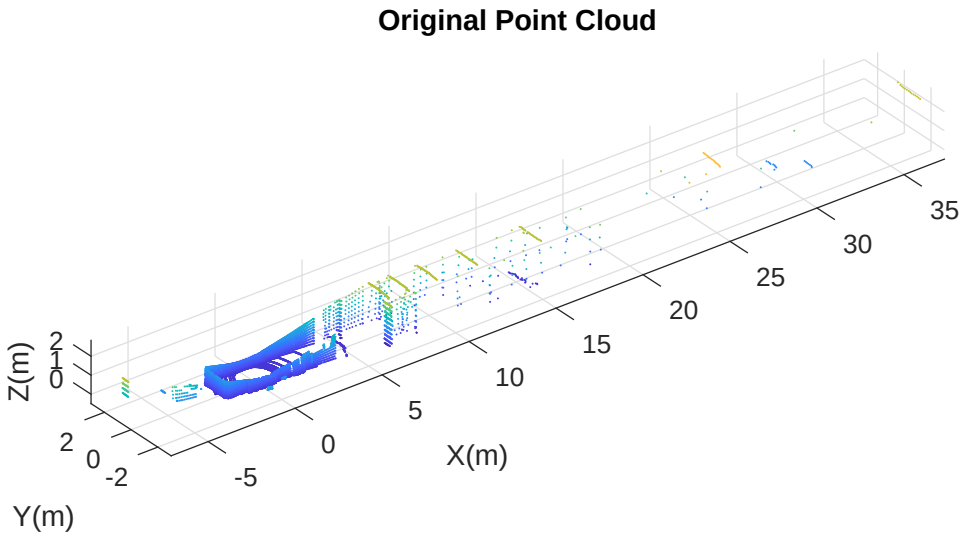


Figure 5.6: Original Velodyne LiDAR point cloud

The point cloud shown in Figure 5.6 was then subject to three sets of transformations to simulate motion in 3-DOF. The order of transformation consists of a rotation around the X axis, a rotation around the Y-axis and then a translational change on the Z axis. The ground plane extraction and isolation demonstration framework are shown in Figure 5.7.

Each stage of the framework shown in Figure 5.7 is then demonstrated below for rotation in 1D.

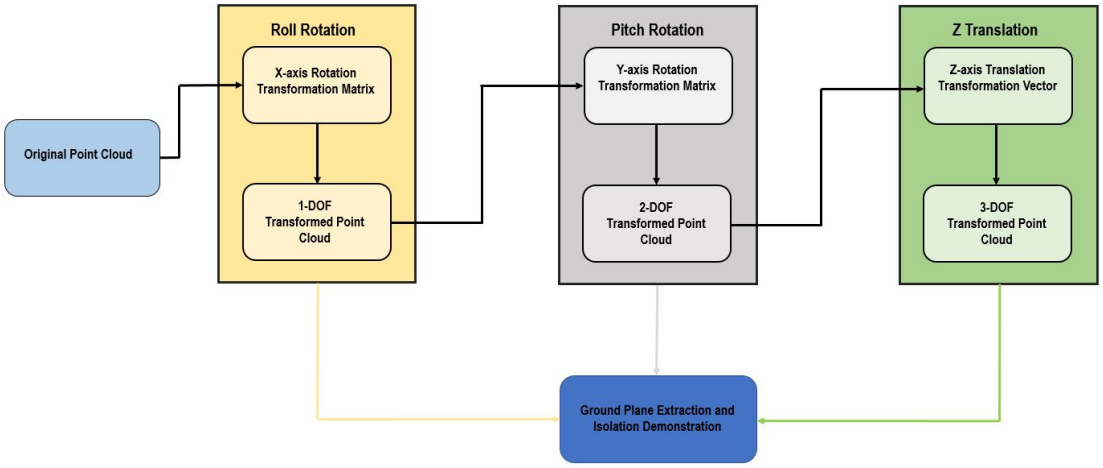


Figure 5.7: Framework for ground plane extraction and isolation demonstration

5.3.1.1 Ground Plane Alignment Example

This section presents a ground plane extraction and isolation demonstration. The original Velodyne point cloud is subjected to a rotation transformation on the X axis according to the rotation matrix ($R_x(\theta)$) shown in Equation (5.4). The angle applied though Equation (5.4) for this demonstration was $\theta = 10°$, this angle was chosen primarily to provide a visually observable change in angle between the extracted ground planes.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\theta) & s(\theta) \\ 0 & -s(\theta) & c(\theta) \end{bmatrix} \tag{5.4}$$

Where;

- c - Cosine
- s - Sine
- θ - Roll angle

The rotation matrix $R_x(\theta)$ was then applied to the original Velodyne point cloud to generate transformed point cloud $\boldsymbol{Q}$. The point cloud $\boldsymbol{Q}$ was then plotted alongside an untransformed version of the original point cloud which is denoted as point cloud $\boldsymbol{P}$. Both $\boldsymbol{P}$ and $\boldsymbol{Q}$ are shown in Figure 5.8 and Figure 5.9 where the rotational offset is evident. Figure 5.8 and Figure 5.9 denote both point clouds as frames, this is an alternative method

of referring to a Velodyne point cloud and is commonly used in literature.

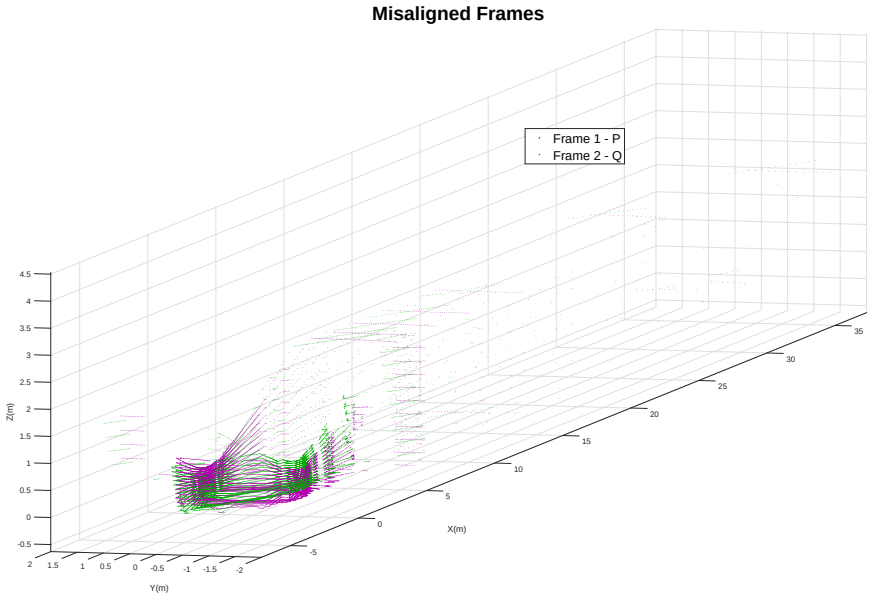


Figure 5.8: Roll misalignment View 1

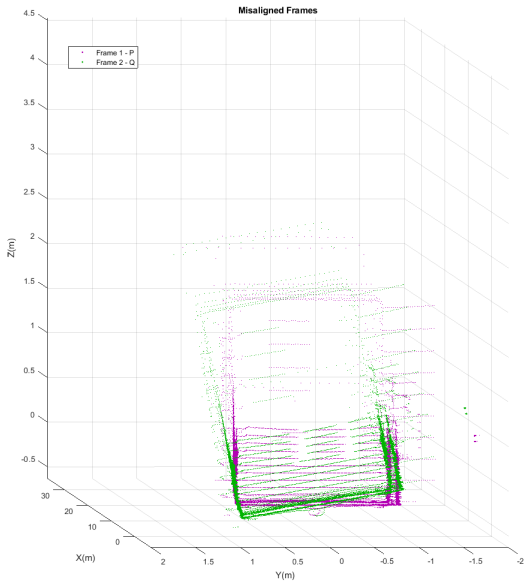


Figure 5.9: Roll misalignment View 2

Figure 5.8 and Figure 5.9 show a misalignment between frames $\boldsymbol{P}$ and $\boldsymbol{Q}$ simulating possible rotational motion that the LiDAR may experience during deployment. The next stage of the demonstration shows the results from the application of the RANSAC plane identification algorithm. Even though the nature of the acquired point cloud frame from

the Velodyne LiDAR is low resolution and sparse, RANSAC detects the sections of the point cloud that represent the dominant ground plane, therefore a planar model can then be applied to enable a clearer representation of the ground plane. This is demonstrated in Figure 5.10 and Figure 5.11.

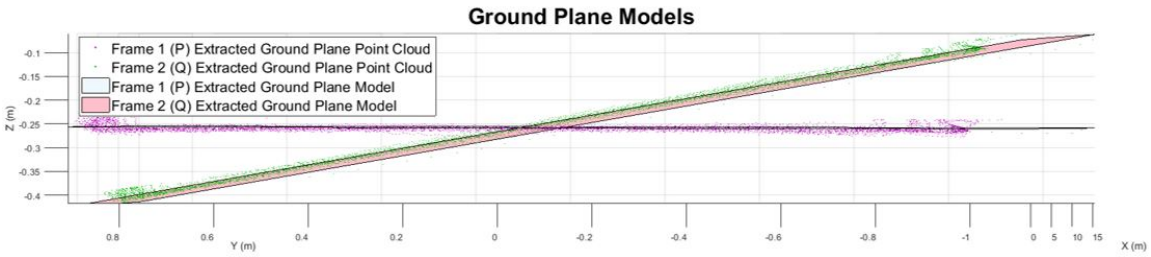


Figure 5.10: Roll ground plane isolation View 1

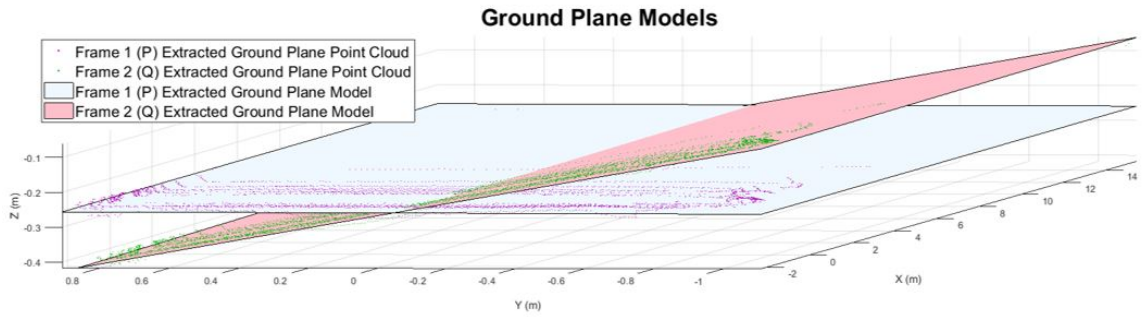


Figure 5.11: Roll ground plane isolation View 2

The results shown in Figure 5.10 and Figure 5.11 observe the output from the RANSAC plane detection algorithm. It shows how the points within the point cloud that represent the ground plane are isolated and a planar model is applied. The misalignment between the two ground planes is also more perceivable. The figures presented in this section show the ground plane isolation and extraction functions when there are two misaligned point clouds as a result of a rotation in 1-DOF. However, it is to be noted that rotation occurrence during active deployment will not be limited to 1-DOF, this is purely for demonstration purposes.

5.3.1.2 Ground Plane Alignment Formulation

Following on from the previous section an introduction into the RANSAC method that was implemented to extract the dominant ground plane, this section analyses how to utilise the extracted ground planes to estimate 6DOF motion. During deployment, if the LiDAR experiences a change in motion between sequential point cloud frames and

this motion is unaccounted for, there will be noticeable motion distortion during the map generation process. Therefore, accounting for such motion is an influential factor in order to design an efficient and effective 3D map generation algorithm. Chapter 4 introduced the initial approaches to solving the 2D localisation problem where an R-EKF was deployed in conjunction with a network of UWB transceivers as a means to provide positioning, using beacons with known locations. The R-EKF was seen to provide improved position estimations in comparison to the native UWB positioning algorithm, however, when placed in unpredictable environments where physical barriers intercepted the signal between beacon and robot, the system struggled to operate effectively. This is what led to the decision to utilise the LiDAR for both 3D mapping and positioning as it enabled the ability to estimate states and motion in 6DOF. Additionally, it was not dependent upon static reference beacons within the environment which would need to be configured prior to deployment. The section of the LiDAR odometry algorithm that uses the extracted ground planes is presented below.

As the LiDAR is subjected to motion change, this results in the misalignment of sequential point cloud frames and further leads to motion distortion in the final generated 3D map. However, it is entirely plausible to use these misaligned frames to estimate motion. The first is to estimate the rotation matrix $\boldsymbol{R}$ that would need to be applied to align the extracted ground planes from scans $\boldsymbol{P}$ and $\boldsymbol{Q}$. The rotation matrix $\boldsymbol{R}$, is computed using the Rodrigues formulation [72]. The Rodrigues rotation formula achieves the rotation of a vector in 3D space, given an axis $\hat{k}$ and angle Θ of rotation [103]. From here, it can be implemented to compute a rotation matrix $\boldsymbol{R}$ in rotation group SO(3), given prior knowledge of $\hat{k}$ and Θ. This can be seen in Equation (5.5).

$$R = I + (sin(\Theta))K + (1 - cos(\Theta))K^2 \tag{5.5}$$

Where;

- I - Identity Matrix
- Θ - Magnitude of rotation (Axis-Angle representation)
- K - Cross product matrix that defines the cross product of any vector $\mathbf{v}$ with $\hat{k}$

The cross product matrix K is represented by the components of $\hat{k}$ and is shown in

Equation (5.6).

$$K = \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix} \tag{5.6}$$

Where;

- $\hat{k}$ - Axis of rotation (Axis-Angle representation)

However, prior to the calculation of $\boldsymbol{R}$, the axis-angle representation of the misalignment between the extracted ground planes from $\boldsymbol{P}$ and $\boldsymbol{Q}$ needs to be computed. The misalignment between the two ground planes can be represented by the misalignment of the unit normal vectors of each plane ($\hat{N}_p$ and $\hat{N}_q$). The optimal rotation matrix is the one that rotates $\hat{N}_q$ by angle Θ about the axis that is perpendicular to the plane containing both $\hat{N}_p$ and $\hat{N}_q$. This perpendicular axis of rotation is represented by $\hat{k}$. Both Θ and $\hat{k}$ are calculated by Equation (5.7) and Equation (5.8). respectively.

$$\Theta = \cos^{-1}(\hat{N}_p \cdot \hat{N}_q) \tag{5.7}$$

$$\hat{k} = (\hat{N}_p \times \hat{N}_q) \tag{5.8}$$

- $\hat{N}_p$ - Normal of ground plane from scan $\boldsymbol{P}$
- $\hat{N}_q$ - Normal of ground plane from scan $\boldsymbol{Q}$

The calculated Θ and $\hat{k}$ are what is known as the Euler angle axis which signifies the Euler rotation theorem. Therefore, it is possible to calculate the roll (θ) and pitch (ϕ) misalignment between $\boldsymbol{P}$ and $\boldsymbol{Q}$ and estimate the motion of the LiDAR.

The change in local altitude between $\boldsymbol{P}$ and $\boldsymbol{Q}$ is the final component which can be calculated from the extracted ground planes. The section of points within $\boldsymbol{P}$ and $\boldsymbol{Q}$ that represent the ground plane are taken and the centroid of each is calculated as c_p and c_q. From this, the translation in the Z direction t_z can be calculated through Equation (5.9) [72].

$$t_z = \frac{c_p}{Rc_q} \tag{5.9}$$

5.3.1.3 Conclusion

With the formulation of the algorithm presented, two different types of testing procedures were then designed to study the algorithm's effectiveness and resulting measurement outputs. The first procedure included a static based experiment, where the mobile LiDAR was left idle to accumulate measurements in order to assess the static drift and noise characteristics. The decision to initially assess the algorithm in static scenarios was to understand how the error behaved prior to embarking upon the planned trajectory. Secondly, a simulation procedure was then designed to test the algorithm's effectiveness and its ability to adjust to varying offsets in angle between two point cloud frames. As the LiDAR is defined as a sensor, the measurements acquired from the device are accompanied with levels of noise which affect its output. Both experimental procedures were designed to model the measurement error and to provide foundational information to enable the construction of systems or methods to handle such error through the application of sensor fusion techniques.

5.3.2 2D Feature Matching

The second part to the proposed two-step LiDAR odometry method consists of a 2D feature matching process. As conducted for the ground plane alignment step, a demonstration is shown to gain insight into how the output from the LiDAR is processed. The feature matching procedure takes a raw 2D laser scan acquired from the Velodyne LiDAR system and applies a sequence of transformations to demonstrate how the feature matching process accounts for motion between two point cloud frames, referred to again as $\boldsymbol{P}$ and $\boldsymbol{Q}$. This process is achieved through the deployment of the Normal Distribution Transformation (NDT) [104]. The overview of the demonstration is shown in Figure 5.12.

5.3.2.1 2D Feature Matching Example

Section 5.3.2.1 presents an overview demonstration of what the data outputted from the LiDAR system experiences when motion occurs between two sequential point cloud frames. This motion simulation is shown in Figure 5.13, more specifically in Figure 5.13a and Figure 5.13b showing a line and point representation respectively. Frame $\boldsymbol{Q}$ shown in Figure 5.13 was simulated to experience a change in motion of 1 metre in both the x and y directions. This value was chosen just to show a clear difference between frames $\boldsymbol{P}$ and

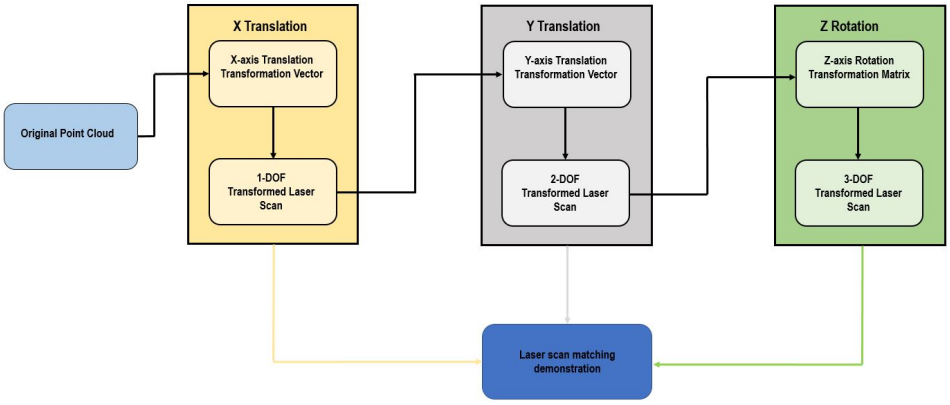
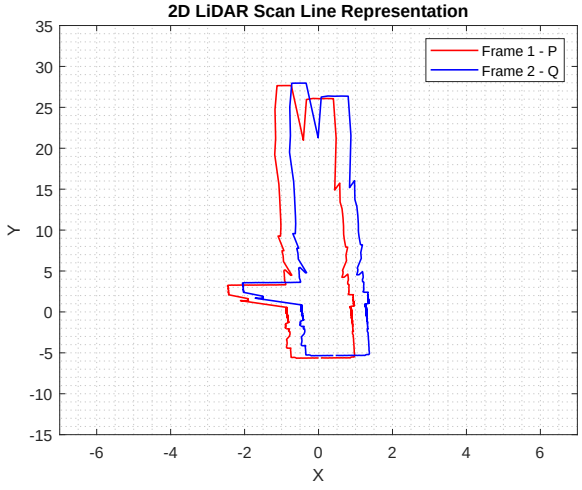
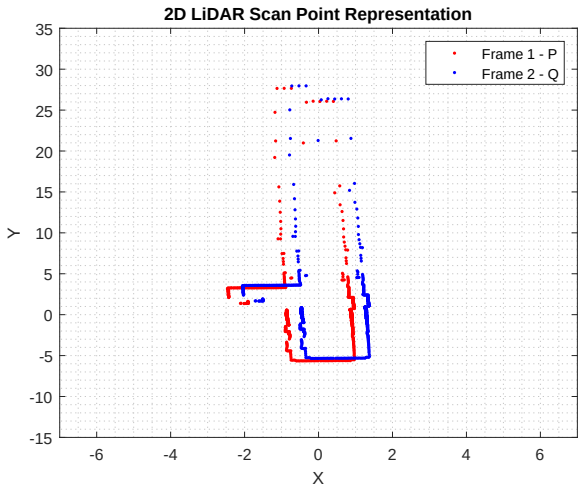


Figure 5.12: 2D laser scan matching demonstration framework

$\boldsymbol{Q}$ in Figure 5.13.



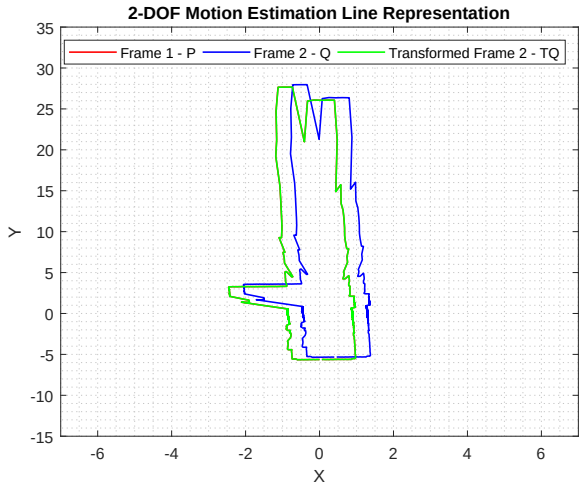
(a)



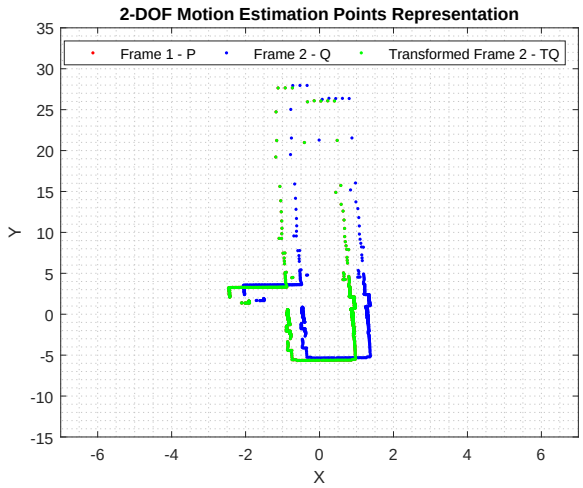
(b)

Figure 5.13: 2-DOF X-Y Translational Misalignment (a) Line representation (b) Points representation

The 2D feature matching algorithm was set to operate using frames $\boldsymbol{P}$ and $\boldsymbol{Q}$ as inputs. The resulting estimation of motion between frames $\boldsymbol{P}$ and $\mathbf{Q}$ was then applied to frame $\boldsymbol{Q}$ as a transformation parameter. Figure 5.14 shows the output of the transformed frame TQ which fully covers frame P. This provides an example of motion estimation between frames P and Q to enable the transformation frame TQ to be aligned with frame P.



(a)



(b)

Figure 5.14: 2-DOF X-Y Translational Misalignment with Transformed Scan (a) Line representation (b) Points representation

5.3.3 Conclusions

In conclusion, Section 5.3 presented an overview of the two-step LiDAR odometry algorithm. It demonstrated through both mathematical implementations as well as graphical simulations of how motion estimations from a dynamic ground-based platform are extracted from raw point clouds acquired from an active LiDAR unit. The following

section presents the experimental procedures deployed to quantify the uncertainty within motion estimations from the algorithm presented in Section 5.3. This step is required in designing robotic control systems and sensor fusion algorithms tasked with estimating motion and tracking the states of an unmanned system.

5.4 LiDAR Ego-motion Uncertainty Characterisation

Following on from the operational procedure demonstrations concerning the 6DOF LiDAR motion estimation algorithm shown in Section 5.3, the work in this section will now present the methodologies undertaken in order to characterise the measurement errors. Section 5.4 is split into two sections labelled Section 5.4.1 and Section 5.4.2 which provide the methodological approaches to quantifying the error for each component of 6DOF motion estimation generated from the LiDAR based algorithm.

5.4.1 2D Feature Matching

Section 5.4.1 presents the structure of the methodologies and presents the results concerning the error characterisation for the components of motion provided by the 2D feature matching stage of the LiDAR algorithm. These components of motion are represented by x, y and ψ denoting Cartesian co-ordinates on the x and y axis and the heading respectively.

5.4.1.1 Static Uncertainty Characterisation Methodology

For the case of motion estimation error characterisation for motion components x, y and ψ, a method that saw the unmanned system operate in a static scenario was applied. The reason behind the choice to operate this experiment statically rather than dynamically was due to the ability to control the error. Due to the nature of the LiDAR, if a dynamic error characterisation approach was undertaken, the error may react differently in relation to the environment such as aspects concerning low feature resolution, laser reflectivity issues and dynamic objects in the environment. Whereas, having a static approach provides consistency in the environment, therefore enabling the quantification of error for each measurement acquired from the LiDAR odometry algorithm only without influence from any external sources. The unmanned system was situated in an indoor environment with no

actuator input so it remained stationary. The feature matching process was set to operate on consecutive point cloud frames and the estimated motion calculated for each iteration was logged as measurement error. The experimental framework can be seen in Figure 5.15. In terms of the 2D scan matching technique, this is its typical operational procedure when deployed on-board the mobile robotic system.

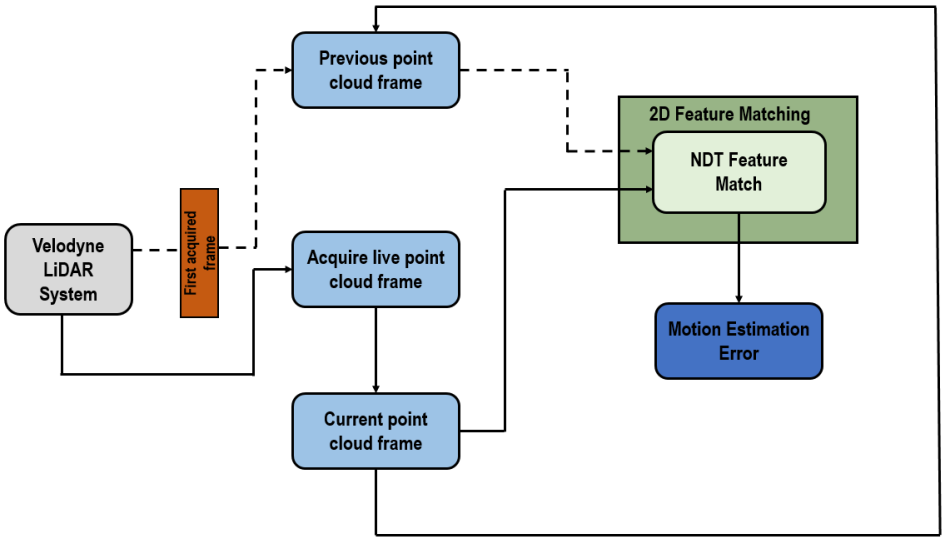


Figure 5.15: Framework for 2D Feature Matching static drift and noise characterisation

The aim of this experiment was to study the error behaviour for each update acquired from the LiDAR. Therefore, the system was left in a static position within an indoor environment to acquire approximately 10,000 LiDAR observations. With the LiDAR operating at an update rate of 10Hz, this resulted in an experimental time of appropriately 1,000s. This enabled the accumulation of a large amount of error data in order to characterise and analyse its nature. An image of the experimental environment is shown in Figure 5.16, the same environment used for the ground plane characterisation methodology presented later in this Chapter. Results concerning this procedure are presented in Section 5.4.1.2.

5.4.1.2 Static Uncertainty Characterisation Results

Section 5.4.1.2 presents the results gathered from executing the experimental process shown in Section 5.4.1.1. It begins with showing the total drift experienced for each component of motion over the total experimental time period. Figure 5.17a, Figure 5.17b and Figure 5.17c represent the static temporal drift along x, y and ψ respectively. It can be seen from these

Figure 5.16: Static experiment environment for 2D scan matching technique

results that with the translational drift on the x and y planes, that the total drift is seen to experience a temporal increase of a linear nature, accumulating to approximately 1.1m and 1m for x and y respectively. The output acquired concerning the drift in x and y demonstrates a similar behaviour to gyroscopic drift in static conditions, however, on a smaller scale. This indicates a possible range error present with the LiDAR system itself, as motion estimation through feature matching produces slight error per iteration. The results acquired for the static based ψ drift shown in Figure 5.17c show significant drift to approximately 20° over the 15 minute time period. This averages out at approximately 0.02° per second, something which the fusion with external sensors may be able to handle.

The results acquired from the static based error characterisation process were then broken down into individual measurement error estimations for each singular iteration. The update frequency for the LiDAR was 10Hz and this process enabled the acquisition of the LiDAR's individual measurement variance for each component of motion. Figure 5.18 presents the generated measurement error statistics for each component of motion provided by the scan matching process. Firstly, Figure 5.18a, Figure 5.18c and Figure 5.18e present the nature of the measurement error over the elapsed time period of the static characterisation procedure. It is noticeable that the measurement estimation errors are produced on a millimetre scale for all three components of motion, where the translational error on the y axis is accompanied with a slight bias yet a lower standard deviation. The results presented along-side the aforementioned errors plots located in Figure 5.18b,

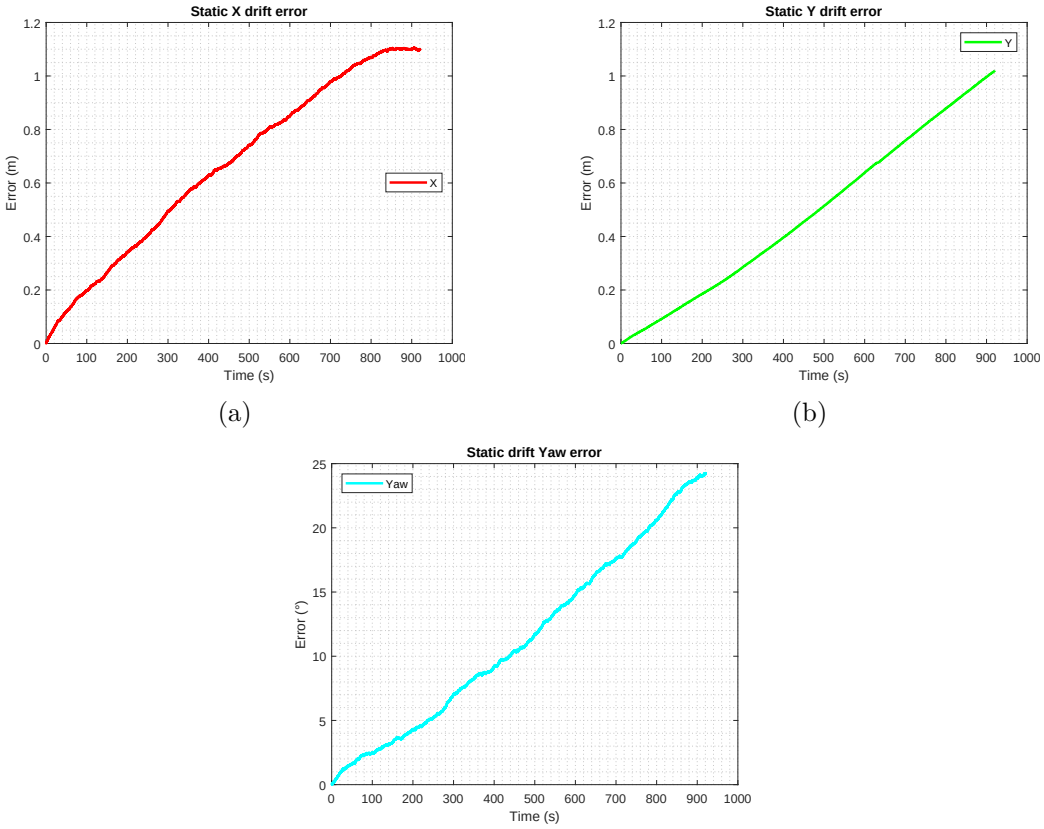


Figure 5.17: Drift over time (a) Displacement X (b) Displacement Y (c) Angular Displacement Yaw (Heading)

Figure 5.18d and Figure 5.18f show the error modelled as a normal distribution for components x, y and ψ respectively. Modelling the error as a normal distribution is a de-facto requirement in the design of an EKF. It is noticeable from the histogram frequency graphics portrayed in Figure 5.18b, Figure 5.18d and Figure 5.18f that the error for each component forms a Gaussian distribution with an appropriate mean of 0 for translational component x and ψ.

Finally, the overall error statistics for the static based error characterisation process is shown in Table 5.1, where the mean μ and standard deviation σ are presented for each axis along with the 95% confidence region. The 95% confidence region is acquired from the normal distribution plot. It states that the generated distribution is 95% confident that either μ or σ lay within their corresponding regions presented in Table 5.1.

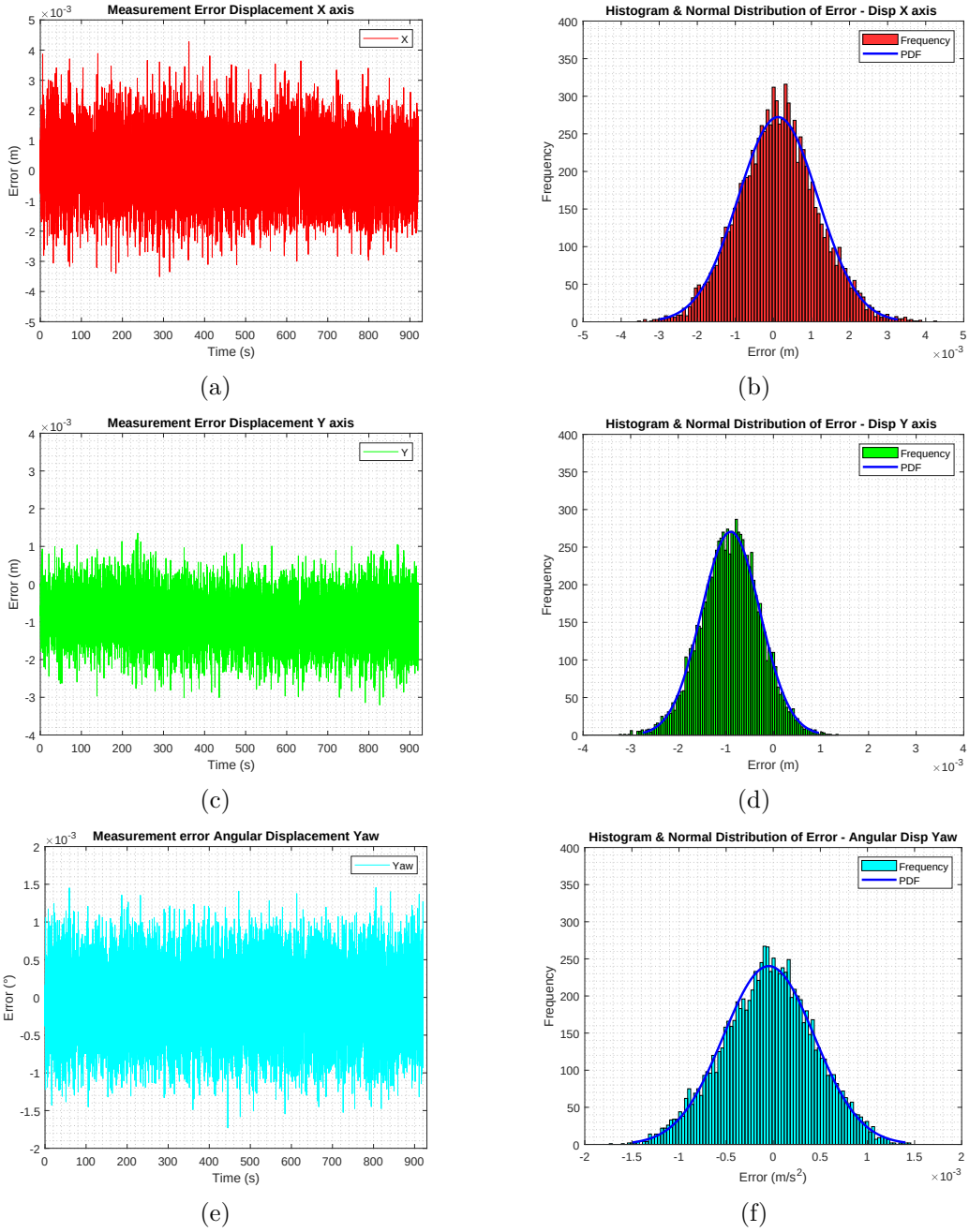


Figure 5.18: Drift over time (a) Displacement X (b) Displacement Y (c) Angular Displacement Yaw

Table 5.1: Static Error Statistics for Components X, Y and Heading (ψ)

Component	$\boldsymbol{\mu}$	$\boldsymbol{\mu}$ **95% CR**	$\boldsymbol{\sigma}$	$\boldsymbol{\sigma}$ **95% CR**
X (m) ($\times 10^{-3}$)	0.121	0.09 — 0.143	1.047	1.032 — 1.062
Y (m) ($\times 10^{-3}$)	-0.898	−0.913 — −0.885	0.617	0.608 — 0.626
ψ (°) ($\times 10^{-3}$)	-0.0465	−0.0564 — −0.0366	0.483	0.476 — 0.490

5.4.1.3 Conclusion

In conclusion, Section 5.4.1 presented the error characterisation methodology and corresponding findings for the static based assessment of the 2D feature matching stage of the algorithm. The results acquired state that the error accompanied with each observation concerning the translational x and y displacement as well as the heading produced low scaled errors by order of millimetres and sub-degree respectively. Over time, however, when remaining in static scenarios, the drift experienced from the Cartesian displacement observations yielded an average of 1.2mm/s. The drift produced in static conditions over the testing procedure time period for the heading was slightly larger than the Cartesian components, producing an error of 0.02°/s. On the whole, the error within each observation generated a Gaussian distribution with relatively low standard deviations. These values are now taken and utilised into the 6DOF-EKF fitting the requirement for Gaussian observation noise matrix.

5.4.2 Dominant Ground Plane Extraction & Alignment

Following on from Section 5.4.1.2 which presented the error characterisation methodology and results for the translational x, translational y and ψ components of motion acquired from the scan matching process. Section 5.4.2 follows a similar approach for the ground plane extraction and alignment stage which estimates the remaining two components of 3D orientation which are roll and pitch, denoted as θ and ϕ respectively and the translational estimate in the z axis. However, Section 5.4.2 presents both static and dynamic error characterisation methodologies, where the latter is in the form of a controlled simulation.

5.4.2.1 Static Uncertainty Characterisation Methodology

Section 5.4.2.1 presents the static methodology conducted in order to characterise and assess the error produced from the ground plane extraction and alignment technique. Motion components concerning the angular displacement in roll and pitch angles as well as estimations of the linear displacement in the Cartesian Z position. The structure of the experimental process conducted can be seen in Figure 5.19.

The process consists of acquiring recursive point cloud frames from the Velodyne LIDAR system and initially identifying the dominant ground plane in each frame. A mathematical implementation is conducted to find the rotational misalignment between the

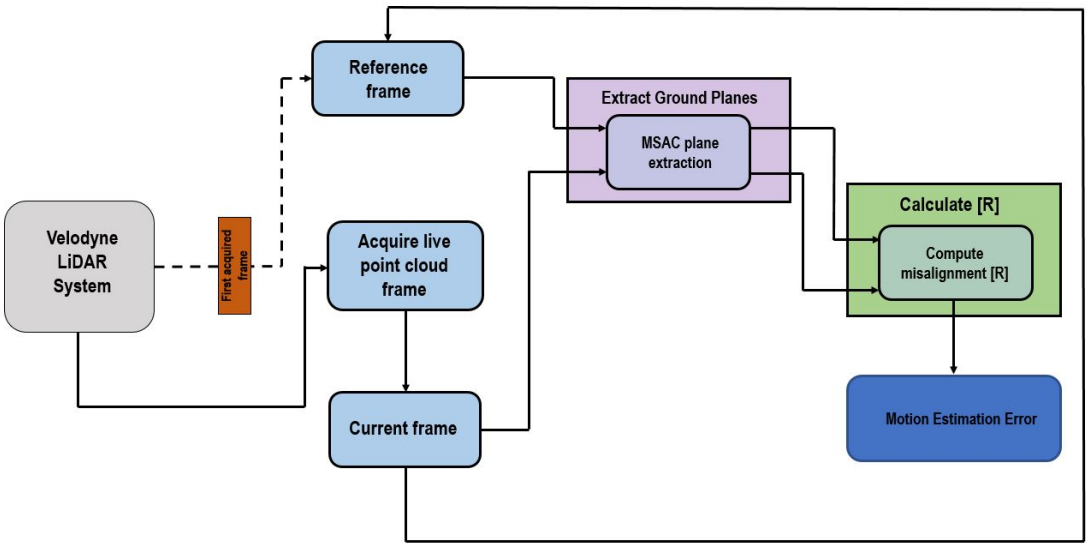


Figure 5.19: Framework for static drift and noise characterisation

two frames. This experimental procedure was carried out in static conditions to provide full controllability of the ground truth metric and to finely estimate the error within each individual observation. The mobile unmanned system was set to idle in a static position within an indoor environment to acquire approximately 10,000 LiDAR observations. With the LiDAR operating at an update rate of 10Hz, this resulted in an experimental time of appropriately 1,000s. The following section presents the results gathered from this procedure.

5.4.2.2 Static Uncertainty Characterisation Results

Section 5.4.2.2 presents the results acquired from the experimental process conducted in Section 5.4.2.1. Firstly, the static drift over the total experimental time period was assessed. These metrics are shown in Figure 5.20, more specifically in Figure 5.20a, Figure 5.20b and Figure 5.20c for roll angle, pitch angle and translation in Cartesian Z respectively. Figure 5.20 displays that the roll and pitch angles showed signs of drift characteristics, where the roll angle was seen to be more volatile. However, the pitch angle was seen to demonstrate signs of a random walk process and maintain at an average of approximately 10° total drift error for the majority of the experimental time period. Final drift angle rates for roll and pitch were calculated at 0.032deg/s and 0.0117deg/s respectively for total drift experienced. The translational Z component of motion, however, was seen to also experience an error drift but on a much lower scale. This nature of the translational Z drift

was seen to be more controlled and of a linear behaviour as shown in Figure 5.20c. The drift rate for this component of motion was calculated at $-0.6743/ms$.

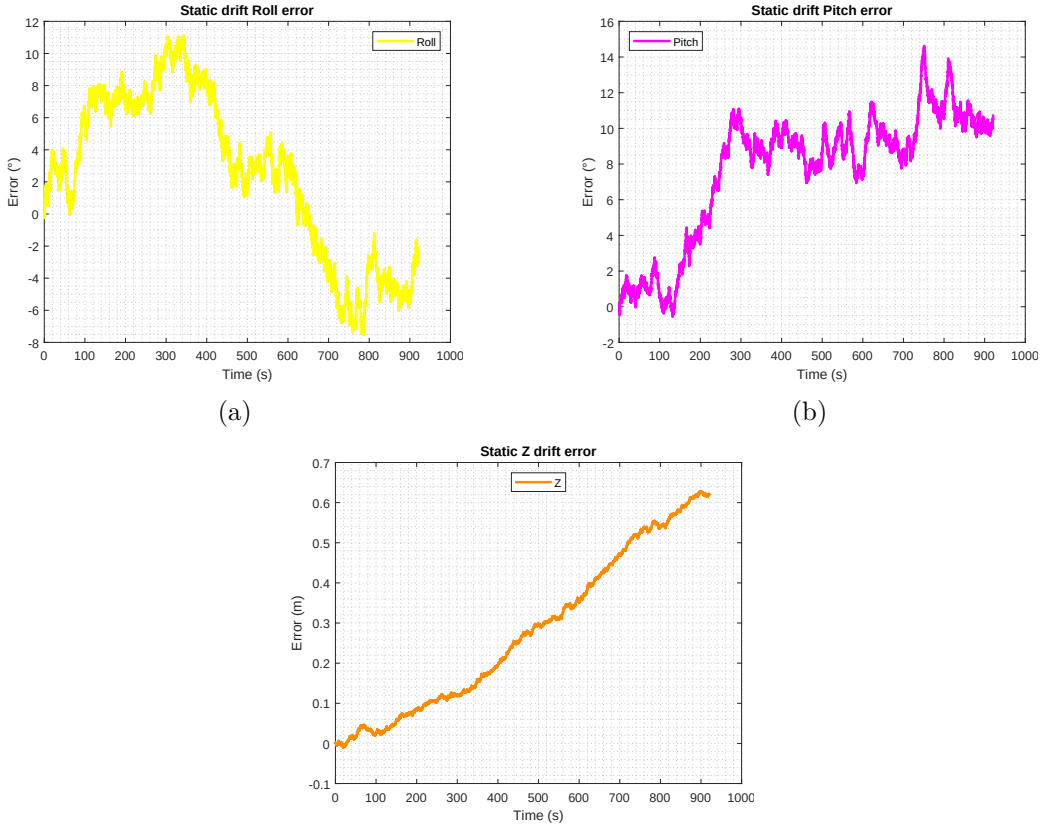


Figure 5.20: Drift over time (a) Angular Displacement Roll (b) Angular Displacement Pitch (c) Displacement Z

Figure 5.20 presented the static drift characteristics acquired from the three components of motion provided through the point cloud ground plane alignment technique. However, an important part in the design of an EKF state estimation algorithm is the noise variance within each individual observation process, this is required to enable the Kalman Gain weighting factor to decide whether to put the trust in the observations more than the process and vice versa. Therefore, an assessment was conducted on each individual observation concerning the three components of motion assessed in this section. These results are seen in Figure 5.21. Firstly, assessing the results concerning the roll angle shown in Figure 5.21a and Figure 5.21b. Figure 5.21a presents the error within each observation over time, which is seen to experience a minimum and maximum range of between $-0.4°$ and $0.4°$. It can be seen that the distribution of error is approximately a zero mean Gaussian distribution, this can further be shown in Figure 5.21b where the error estimation frequencies form the

same bell-shape curve as that of a Gaussian distribution. Similar results were acquired for the pitch angle, which are displayed in Figure 5.21c and Figure 5.21d. However, the errors within each pitch observation were of a smaller scale compared to that of the roll, where it is seen to experience approximate minimum and maximum range of between $-0.2°$ and $0.2°$. This indicates that the pitch angle obtained lower standard deviations of error. The nature of the pitch observation error also took the form of a zero mean Gaussian distribution, this is observable in Figure 5.21d. Finally, analysing the displacement results from the Cartesian Z component showed a lower observation variance in respect to the roll and pitch. Although they differ in terms of angular and linear displacement and the resulting unit representations, the linear displacement observations for the Z component proved to be the most stable. The approximate range of the error is seen to be approximately -2mm - 2mm centering at around a zero mean, this can be seen in Figure 5.21e. Therefore, the resulting error distribution plot in terms of frequency generated a fine Gaussian plot and also demonstrated a zero mean value, however, with a lower standard deviation in comparison to the roll and pitch angles with respect to their corresponding units of measurement. This can be seen in Figure 5.21f. The final results for the static error characterisation procedure for components roll, pitch and Z can be seen in Table 5.2. The mean error of the Z component of motion was minuscule so this value was set to zero.

Table 5.2: Static Error Statistics for Components Roll (θ), Pitch (ϕ) and Z

Component	$\boldsymbol{\mu}$	$\boldsymbol{\mu}$ **95% CR**	$\boldsymbol{\sigma}$	$\boldsymbol{\sigma}$ **95% CR**
θ (°)	-0.0002	$-0.00305 - 0.00246$	0.251	$0.2491 - 0.2530$
ϕ (°)	0.0012	$-0.00065 - 0.00303$	0.101	$0.09 - 0.103$
Z (m) ($\times 10^3$)	0	0	0.869	$0.857 - 0.882$

5.4.2.3 Simulated Dynamic Rotations Methodology

Similar to the previously introduced demonstration procedure, the simulated dynamic rotation testing procedure includes the use of an original point cloud $\boldsymbol{P}$ and a transformed point cloud $\boldsymbol{Q}$. The rotation simulation analysis was designed to enable full control of the applied rotations to the point cloud frames. Deploying the robot live for experimental purposes would be more difficult to achieve the level of controllability to acquire a ground truth metric for error calculation. This is why a simulation was chosen. However, the

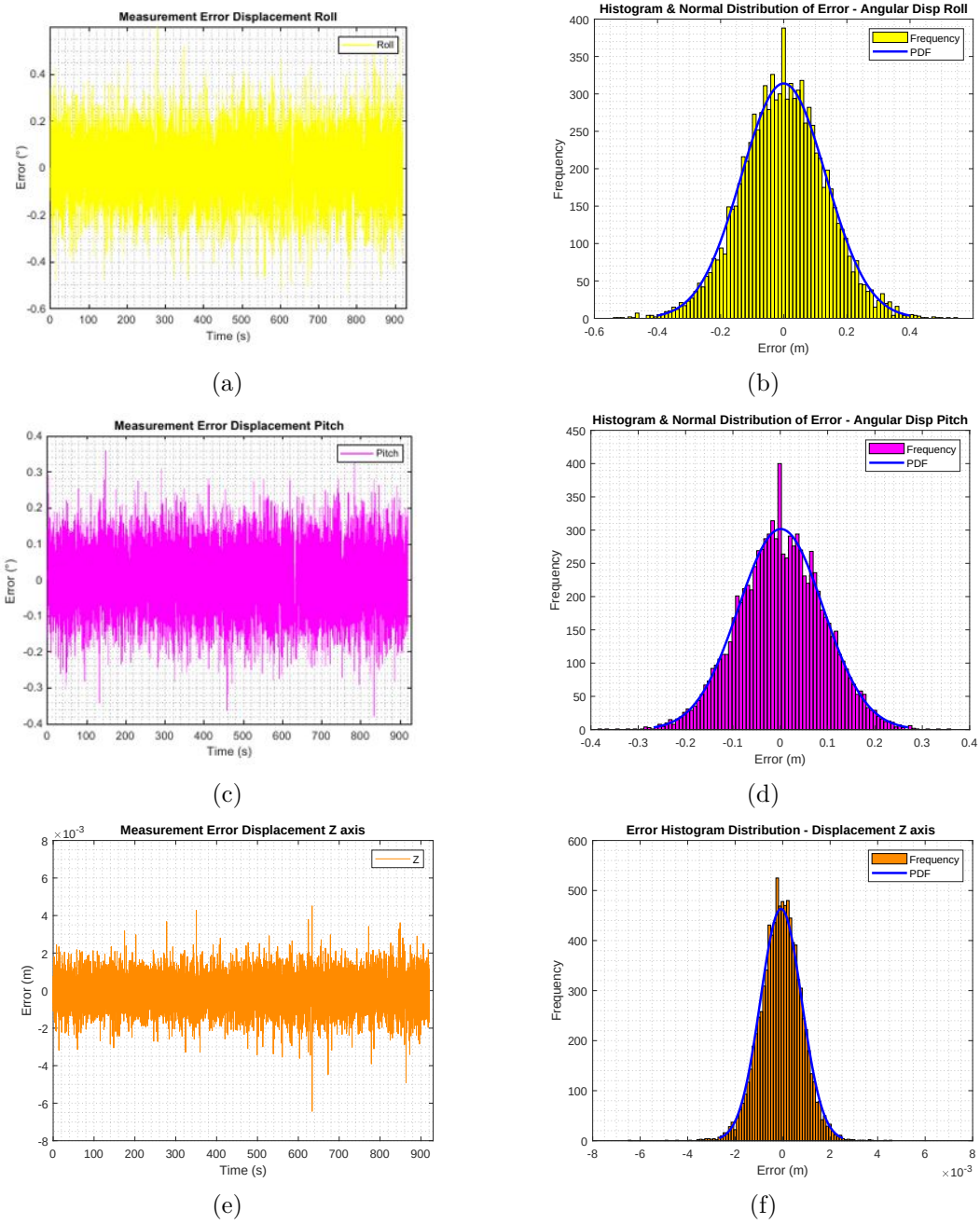


Figure 5.21: Drift over time (a) Displacement X (b) Displacement Y (c) Angular Displacement Yaw

simulation procedure consisted of applying controlled rotations to previously acquired consecutive point cloud from the Veldoyne LiDAR, stating that the point cloud was not synthetically generated. The rotation simulation procedure was carried out on both the roll and pitch axes but in separate instances. The rotation matrices $R(\theta)_x$ and $R(\phi)_y$ were applied to the original point cloud frame $\boldsymbol{P}$ for roll and pitch respectively to generate point cloud $\boldsymbol{Q}$. These rotation matrices are shown in Equation (5.10) and Equation (5.11), where θ and ϕ were incrementally increased by 0.5° per second until the maximum testing angle

of 25° was reached. A small angle increment of 0.5° was chosen to study how the algorithm dealt with low scale changes in angle between $\boldsymbol{P}$ and $\boldsymbol{Q}$. A simulation analysis of the linear displacement concerning the z component was not assessed due to the system being a ground-based vehicle and was assumed not to experience any high changes in relative altitude.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\theta) & s(\theta) \\ 0 & -s(\theta) & c(\theta) \end{bmatrix} \tag{5.10}$$

$$R_y(\phi) = \begin{bmatrix} c(\phi) & 0 & s(\phi) \\ 0 & 1 & 0 \\ -s(\phi) & 0 & c(\phi) \end{bmatrix} \tag{5.11}$$

The simulated dynamic rotation testing framework as a whole is shown in Figure 5.22. The testing procedure included five main iterations of the $0° - 25°$ rotation simulation with ten repetitions for each iteration. The details concerning test repetition are shown in Table 5.3.

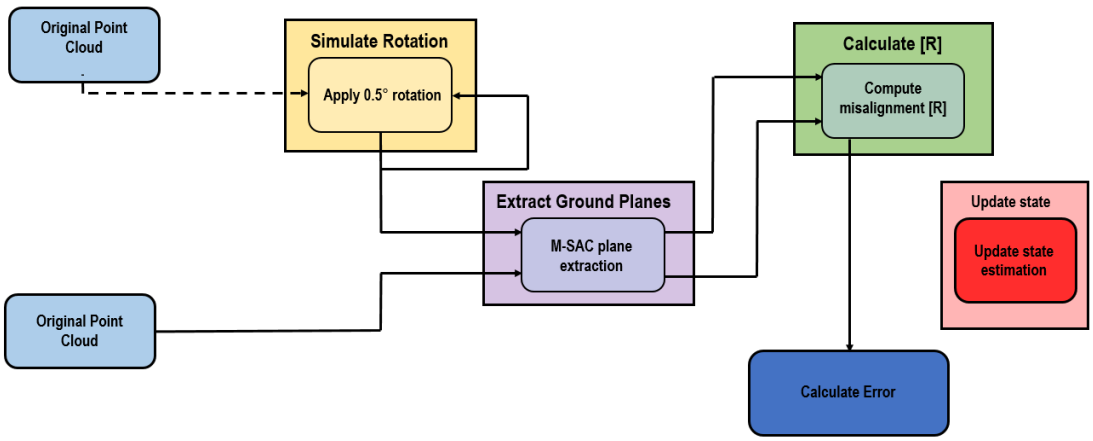


Figure 5.22: Framework for simulated rotation testing

The results achieved from implementing the previously presented methodology are included in the following section.

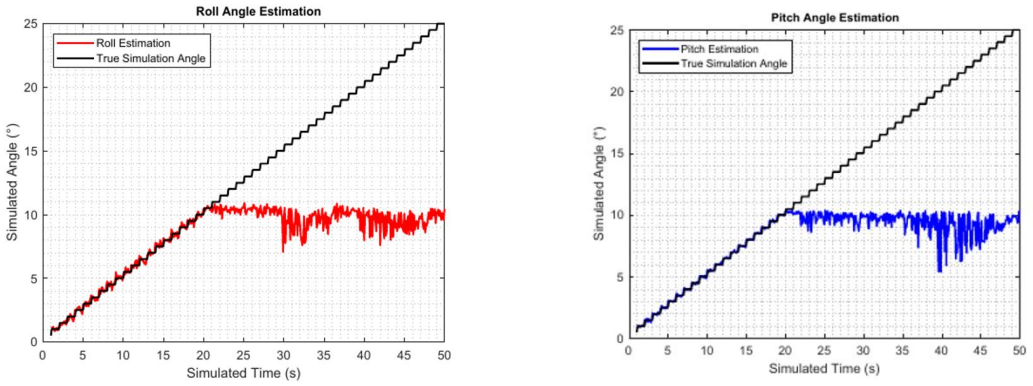
5.4.2.4 Simulated Dynamic Rotations Results

This section presents the results gathered from the implementation of the rotation simulation analysis for both the roll and pitch angles. The initial stages of the rotation

Table 5.3: Rotation Simulation Test Details

Iteration No.	**No. of Repetitions**	**No. of Measurements**
1	10	5000
2	10	5000
3	10	5000
4	10	5000
5	10	5000
Total	**50**	**25,000**

simulation analysis generated unusual results. The application of the Rodrigues formula based algorithm to extract rotation motion from two ground planes was seen to fail as the planes reached a 10.5° misalignment on both the roll and pitch axis. These initial results therefore meant that there was a major limitation in relation to a maximum of 10.5° misalignment angle between two point cloud frames before the algorithm failed and lost track of the current angle. This occurrence is shown in Figure 5.23 for both roll and pitch.



(a) Initial roll angle estimation

(b) Initial pitch angle estimation

Figure 5.23: Initial angle estimation

Considering the results shown in Figure 5.23 there was a clear necessity to overcome a huge limitation found in the initial testing procedures. During deployment and intended environment operational procedures, the mobile system would highly likely encounter scenarios where the roll and pitch components of the robots 6DOF pose would exceed 10.5°. This could be due to gradually inclining or declining terrains or avoiding unpredictable objects within the environment. Therefore, estimation of roll and pitch estimation from utilising this ground plane alignment method was unsuitable.

5.4.2.5 Conclusion

To conclude, Section 5.4.2 presented the static and dynamic error characterisation methodologies and results for the ground plane alignment technique. The alignment of dominant ground planes within two consecutive point clouds provided estimation of motion components roll angle, pitch angle and linear displacement in the Z direction. Within both static and dynamic experimental procedures, the aforementioned components of motion were seen to produce observation errors of a zero mean Gaussian distribution when assessed individually over the course of the experimental procedure. When comparing the results acquired from the static and dynamic experimental procedures for the roll and pitch angles, final error statistics yielded very similar results, demonstrating that in both static and dynamic conditions, observations concerning the roll and pitch angles operate identically. The acquisition of these metrics enabled understanding of the observation error prior to the integration into the design of the EKF.

5.4.3 Conclusion

In conclusion, Section 5.4 presented the error characterisation experimental procedures and results concerning 6DOF motion estimations from point cloud data acquired from a Velodyne VLP-16 system. The motion estimations were separated into two stages, a 2D scan matching technique which estimated displacement in Cartesian X and Y and heading and a dominant ground plane alignment technique, which estimated motion for components roll, pitch and Cartesian displacement in Z. Static based testing procedures concerning the motion output from the 2D scan matching algorithm produced errors on a millimetre scale for observations concerning X and Y and sub-degree errors for heading observations at each time iteration. However, when subjected to static conditions over long periods of time, these estimations were seen to drift. Concerning the ground plane alignment technique, both static and dynamic error characterisation methodologies were conducted to assess the roll and pitch angles. The error metric outputs from the static based procedure showed that there were small error variations produced when the ground plane alignment algorithm gestated roll and pitch change. However, during the simulated dynamic experimentation the algorithm failed once the angle between two sequential ground planes reached 10.5°. This proved that the estimation of motion through ground plane extraction using the

Rodriques rotation formula possesses limitations which needed to be overcome. Therefore, an alternative method was assessed through initially studying the on-board IMU.

5.5 Inertial Measurement Unit

The IMU built in to the Pixhawk FCU mounted in the chassis of the robot was assessed in terms of its static error. As the accelerometer and gyroscopic readings were utilised as a control input in the state estimation process, their uncertainties and biases were also characterised. This enabled the generation of the control input noise matrix, U which accounts for uncertainties of the components used for the input. The control input noise matrix is included in the calculation that predicts the state covariance matrix P. The following sections include an overview into the IMU uncertainty and bias acquisition and the resulting findings.

5.5.1 Static Testing Methodology

The characterisation of the IMU built in the Pixhawk FCU was carried out in a static nature. This enabled the acquisition of an offset bias and the standard deviation of the IMU measurements in relation to the bias. In order to overcome the identified limitation of the ground plane extraction and alignment method through the use of LiDAR, the IMU was initially assessed as a standalone prior to integrating a sub-sensor fusion algorithm. The testing structure consisted of 10 repetitions where the robot has statically positioned on a rigid plane, preventing any possibility of any external vibration occurrence. These tests were operated independently. Table 5.4 shows details concerning the number of measurements and elapsed time for each test.

The results acquired from each iteration were then accumulated to increase overall sample size for the purpose of Gaussian modelling. However, as each test iteration had its own independent time characteristics, it was necessary to acquire a total elapsed time of the whole testing procedure. This enabled an understanding concerning the behaviour of error over time. A graphical representation of the independent time is shown in Figure 5.24a, where it is clear that each test was executed for approximately 200s and resetting back to 0, also previously confirmed in Table 5.4. The accumulation of time, carried out to assess the error trend for each IMU component over the total time period of the 10 test iterations, was

Table 5.4: IMU Static Testing Structure

Test Iteration No.	**No. of Measurements**	**Elapsed Time (s)**
1	10091	201.78
2	9813	196.19
3	10018	200.31
4	10120	202.36
5	10076	201.48
6	10104	202.04
7	10012	200.20
8	10139	202.73
9	10149	202.93
10	10272	205.39
Total	**100794**	**2015.4**

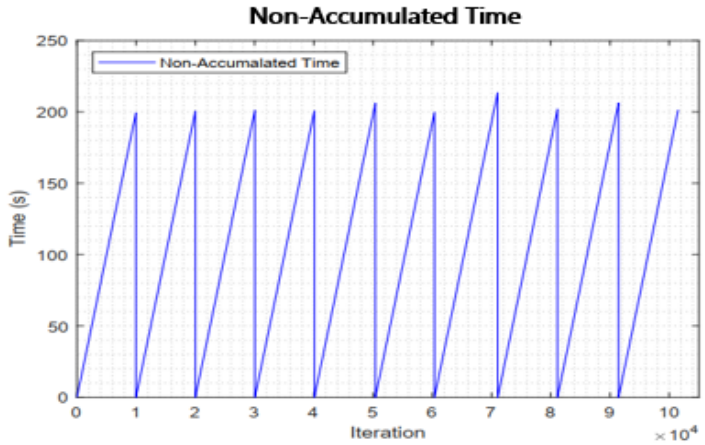
achieved using the final time stamp of the previous test as an initial offset. For example, the first time reading for test iteration 2 was set to the final time reading of test iteration 1. This was repeated for each test iteration. The final accumulation of time is shown in Figure 5.24b, where the linear nature of the plot shows the expected gradual increase in time to approximately 200s, also stated in Table 5.4.

5.5.2 Static Testing Results

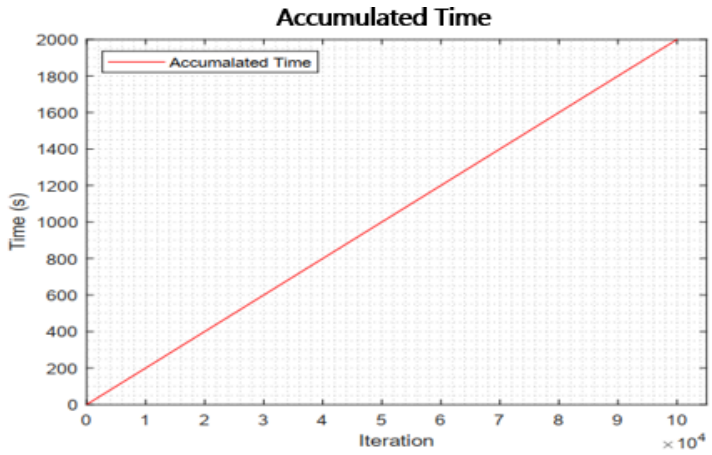
The following sections present the statistics gathered from the implementation of the static IMU uncertainty and bias characterisation methodology.

5.5.2.1 Accelerometer

By nature, accelerometers are known to produce readings accompanied by a form of measurement noise. The rapid operating frequency of accelerometers is an influential factor in the generation of this measurement noise. However, the fast updating characteristic of an accelerometer is beneficial when accounting for periods of time where external measurements are not present, such as within a GNSS/INS system, where the update rate of a GNSS can be once per second. Therefore, a decision was taken to model the output of the accelerometer at its native operational frequency and characterise noise and bias by accumulating large samples of data. Figure 5.25 contains all the error statistics gathered from the static testing



(a)



(b)

Figure 5.24: Time Characteristics (a) Independent Time (b) Accumulated Time

in terms of the behaviour of the noise over time for the accelerometers operating on the X, Y and Z axes. Figure 5.25 is separated into two columns, where the first column represents the error trend over time and the second column demonstrates the error frequency plot in the form of a histogram and the inclusion of a normal distribution to demonstrate error behaviour. This is done for each of the three axes and the results are shown in Figure 5.25.

Results within Figure 5.25a show the modelled error for the X axis. It is clearly observable that post accumulation of the large dataset conducted for the static test shows that the X axis accelerometer possesses an initial static bias. This is demonstrated by the cyan coloured dataset. This bias again is clearly visible when analysing the nature of the error frequency shown in Figure 5.25b, where this is represented through the offset between the two normally distributed error statistics. Figure 5.25a and Figure 5.25b include the error statistics post bias removed, which is represented in both figures by the magenta

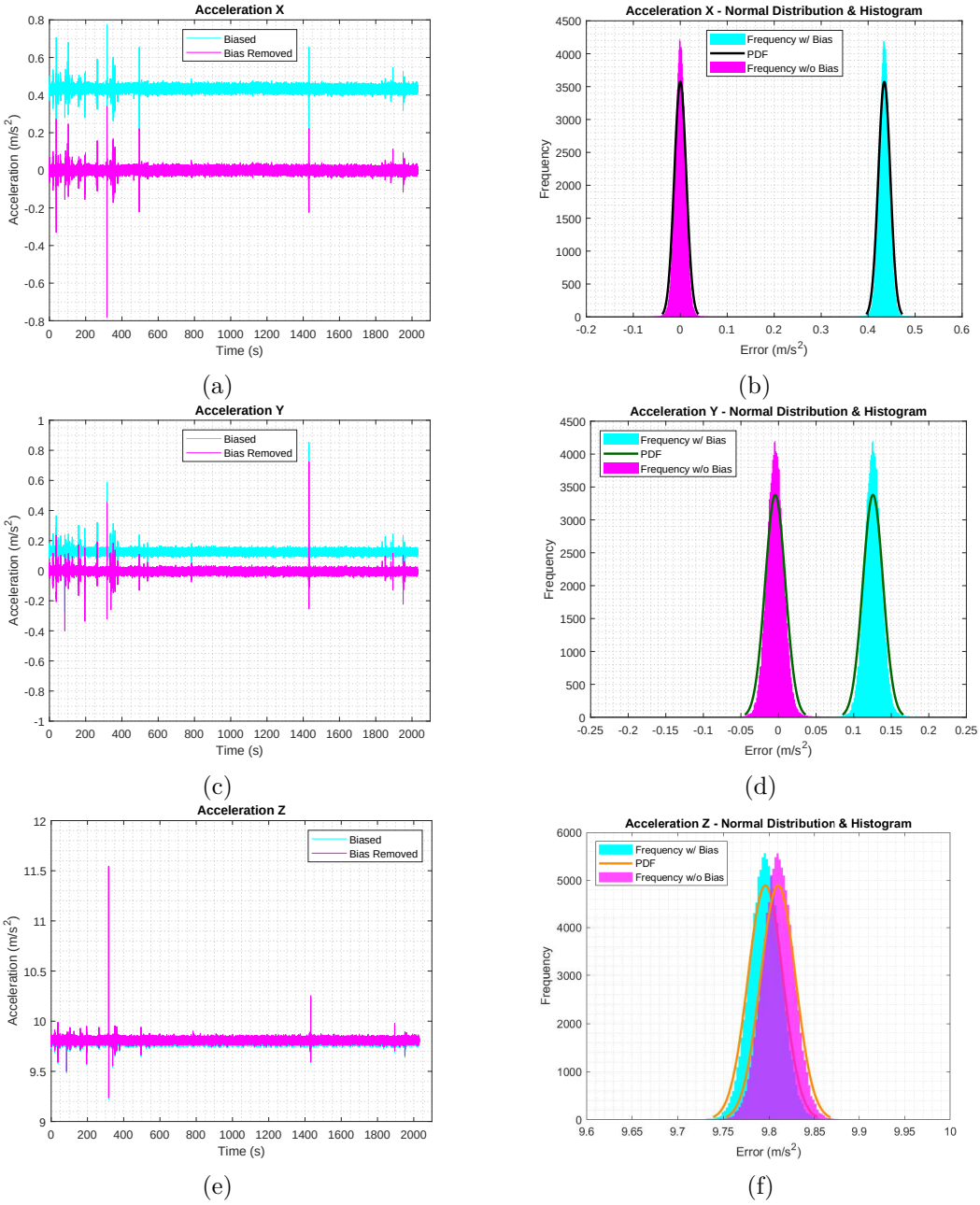


Figure 5.25: Accelerometer Error (a) X axis (b) X axis ND & Histogram (c) Y axis (d) Y axis ND & Histogram (e) Z axis (f) Z axis ND & Histogram

dataset. Accounting and identifying any bias within the measurements enables its removal and further mitigating any effect that the bias may have on future orientation estimations. As a whole, the accelerometer readings for the X axis proved to have a Gaussian distribution of error. This can be seen through observing the shape of the histogram Figure 5.25b.

The results presented in Figure 5.25c show the modelled error for the X axis. The error trend shown in Figure 5.25c is very similar to that of the static measurement error of the accelerometer on the X axis in relation to its behaviour and distribution. However,

Figure 5.25c shows that the accelerometer on the Y axis possesses a lower native bias in comparison to that of the X axis. This can be observed by the identifying the offset concerning the cyan coloured plot shown in Figure 5.25c. The distribution of error frequency for the accelerometer on the Y axis also possesses a Gaussian like shape, which is observable by the histogram charts and the resulting normal PDF in Figure 5.25d.

Figure 5.25e and Figure 5.25f presents the acquired results concerning the Z axis accelerometer. Figure 5.25e provides the details concerning the biased and post bias removal datasets for the Z axis accelerometer. This figure demonstrates the accelerometer initially reading at approximately 9.81m/s^2, which is the force due to the gravitation vector. It can be seen from both Figure 5.25e and Figure 5.25f that the accelerometer output on the Z axis has a slight initial bias as it reads below the expected value of 9.81m/s^2. This can be best observed by the shifted offset between the two produced Gaussian distributions of error within Figure 5.25f.

Finally, overall statistics for the accelerometer error are tabulated in Table 5.5, where the mean μ and standard deviation σ are presented for each axis.

Table 5.5: Accelerometer Error Statistics

Axis	$\boldsymbol{\mu}$	$\boldsymbol{\mu}$ **95% CR**	$\boldsymbol{\sigma}$	$\boldsymbol{\sigma}$ **95% CR**
X (m/s^2)	0.4344	$0.4343 - 0.4345$	0.01275	$0.0127 - 0.01281$
Y (m/s^2)	0.1257	$0.1256 - 0.1258$	0.01347	$0.01344 - 0.0135$
Z (m/s^2)	-0.0139	$-0.0141 - -0.0137$	0.0191	$0.0190 - 0.0192$

5.5.2.2 Gyroscope

Similar to accelerometers, measurements acquired from gyroscopes also include slight biases of a certain degree. Carrying out experimental procedures to optimally estimate a generalised bias variable is a constructive method to undertake to reduce overall uncertainty within the measurements. The results concerning static gyroscope error and bias characterisation for the X, Y, and Z axes are shown in Figure 5.26. Identical to the results structure presented for the accelerometers, Figure 5.26 contains two columns presenting error distribution in relation to time and error frequency distribution for all three axes.

The results acquired for the static error behaviour in relation to time for the gyroscopes

on the X, Y and Z axes represented by Figure 5.26a, Figure 5.26c and Figure 5.26e respectively yield very similar results. Although there are signs of outlier occurrences for the X and Y axes' gyroscopes, possibly caused by external and environmental perturbations, these outliers are almost deemed negligible to the mass quantity of the data acquired in certain static conditions. This however, does not occur for the Z axis gyroscope as shown in Figure 5.26e. The error frequency for each axis represented in Figure 5.26b, Figure 5.26d and Figure 5.26f for X, Y and Z respectively demonstrates almost identical results in terms of over-all distribution, approximate mean and standard deviation of error. It is clear that all gyroscopes possess a native bias which is almost unidentifiable as shown in Figure 5.26a, Figure 5.26c and Figure 5.26e and is therefore not included within the corresponding normal distribution plots. As a whole, the gyroscopic static error output yielded distributions of error that are of a Gaussian nature.

The final results gathered from the gyroscopic error characterisation procedure for mean and standard deviation or error are described below in Table 5.6 along with the corresponding CRs. The values for the mean error (μ) and the 95% CR for the mean within Table 5.6 are measured in units of $\times 10^{-4} deg/s$. Although the gyroscopic bias was very small as shown in Figure 5.26, it is included within Table 5.6 as these values will be integrated into the design of the ES-KF.

Table 5.6: Gyroscope Statistics

Axis	$\boldsymbol{\mu}(\times 10^{-4})$	$\boldsymbol{\mu}$ **95% CR** $(\times 10^{-4})$	$\boldsymbol{\sigma}$	$\boldsymbol{\sigma}$ **95% CR**
X (deg/s)	-1.976	$-6.901 - 2.591$	0.0801	$0.0798 - 0.0805$
Y (deg/s)	-1.552	$-6.662 - 3.549$	0.0831	$0.0827 - 0.0834$
Z (deg/s)	-3.80×10^{-1}	$4.606 - 3.559$	0.0687	$0.0684 - 0.0690$

5.5.3 Conclusions

In conclusion, Section 5.5 presented the error characterisation process, results and analysis for the IMU that operates on-board the unmanned system. The importance behind this characterisation process enables the acquisition of any biases that the sensors within the IMU possess in resting and static conditions. The results gathered from Section 5.5 show that both proprioceptive sensors that were tested form a Gaussian distribution of error when measurements were accumulated over a long period of time. The nature of any

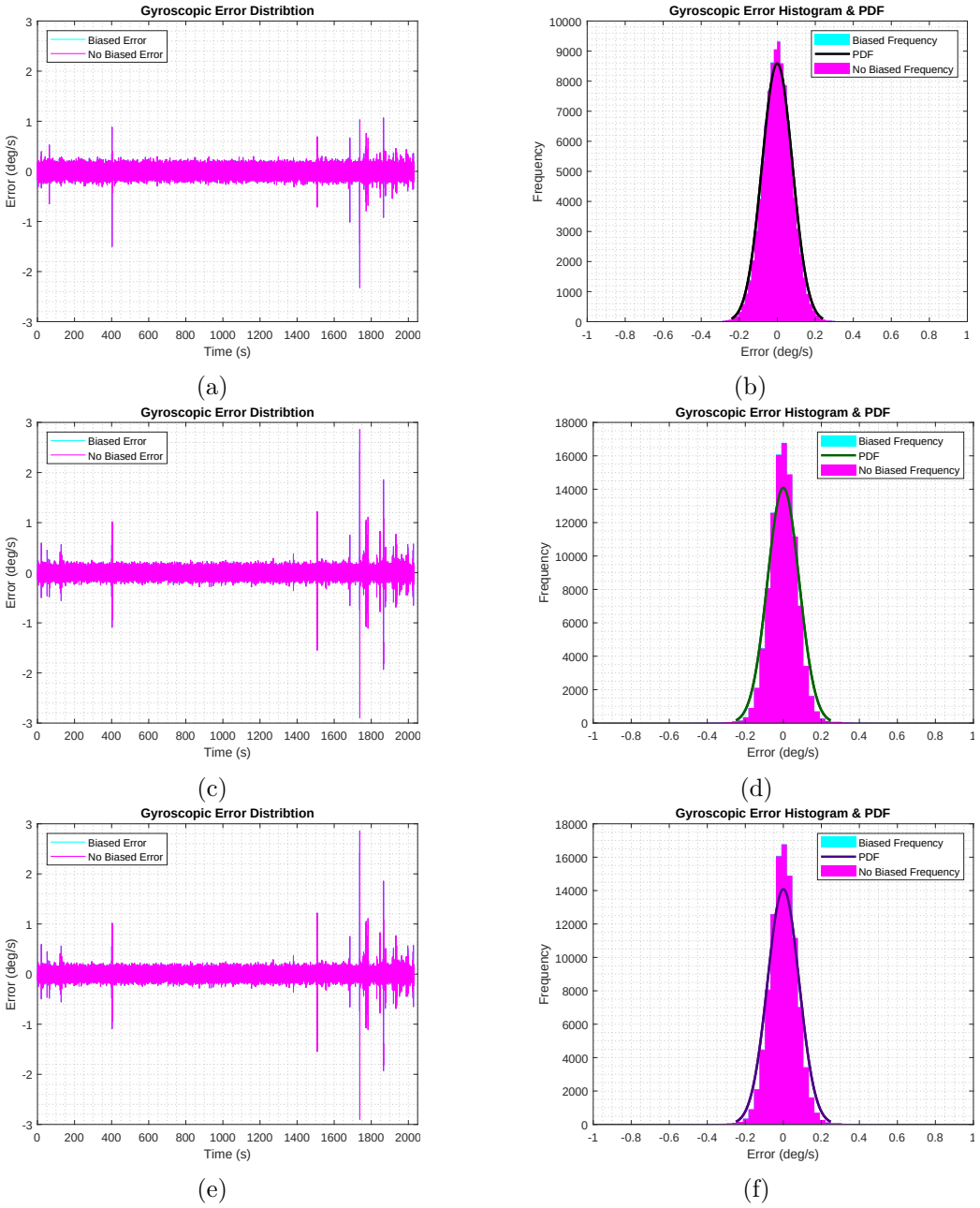


Figure 5.26: Gyroscopic Error (a) X axis (b) X axis ND & Histogram (c) Y axis (d) Y axis ND & Histogram (e) Z axis (f) Z axis ND & Histogram

KF algorithm is that it assumes that sensor measurement covariance is Gaussian, therefore demonstrating that the sensors within IMU follow this assumption and can be implemented into Gaussian based state estimation algorithms for optimal performance. As an alternative method to overcome the angle limitation between two sequential ground planes found in Section 5.4.2.4, the sensors within the IMU are to be used within a sub-state estimation technique known as the Error State Kalman Filter (ES-KF). The ES-KF will act as a sub-process within the 6DOF-EKF providing estimations concerning the roll and pitch

states. These error statistics acquired from the IMU based experimental procedures are now extracted and taken into the design of the ES-KF for orientation calculation from a fusion between three axis accelerometers and gyroscopes. The main values taken from Section 5.5 are outlined below in Equation (5.12), Equation (5.13), Equation (5.14) and Equation (5.15) which represent the three axes' biases (β) and the standard deviation of errors (σ) for the accelerometers and gyroscopes respectively. These values are represented by column vectors.

$$\beta_{\alpha} = [0.434,\ 0.127,\ 0.014]^T \tag{5.12}$$

$$\beta_{\omega_{x,y,z}} = [-1.976,\ -1.552,\ -0.38]^T 10^{-4} \tag{5.13}$$

$$\sigma_{\alpha_{x,y,z}} = [0.0128,\ 0.0135,\ 0.0191]^T \tag{5.14}$$

$$\sigma_{\omega_{x,y,z}} = [0.0801,\ 0.0831,\ 0.0687]^T \tag{5.15}$$

The integration of the values within Equation (5.12), Equation (5.13), Equation (5.14) and Equation (5.15) are now implemented in the design of the ES-KF within Section 5.6.

5.6 Error State Orientation Kalman Filter

An important component in any robotic system is the IMU which in this case, consists of 3 axis accelerometers and gyroscopes providing linear acceleration and angular velocity respectively. This configuration within an IMU is known as a 6 axis representation and when combined with a 3 axis magnetometer, totals to a 9 axis IMU representation. In this case, the ES-KF will provide observations to the 6DOF-EKF through a fusion of the accelerometers and gyroscopes complementing the limitations of each individual sensory system.

5.6.1 Orientation from IMU

Although accelerometers provide measurements concerning the linear acceleration along each axis, they can also be utilised in conjunction with gyroscopic information to estimate the orientation of the gravitational vector. The application of this gravitation vector calculation can therefore enable the estimation of the orientation of the platform that the IMU is mounted upon, in this case, a mobile ground-based robot. However, accelerometers

and gyroscopes as stand-alone systems can be used to provide estimations concerning orientation, with each system possessing limitations in its estimations. The mathematics in order to calculate orientation from accelerometers and gyroscopes as stand-alone systems are outlined in Section 5.6.1.1 and Section 5.6.1.2 respectively. Additionally, the example output from the orientation calculation from each sensor, based on the static dataset acquired for the accelerometer and gyroscope bias and variance characterisation methodology presented in Section 5.5 is also included within Section 5.6.1.1 and Section 5.6.1.2.

5.6.1.1 Accelerometer

The mathematical implementation to extract orientation from accelerometers is finding the angles on the X and Y axis that the gravitational vector is aligned to in an instance in time. This is achieved for the roll (θ) and pitch (ϕ) angles by Equation (5.16) and Equation (5.17) through the application of the four quadrant inverse tangent. It is to be noted that the accelerometer is unable to estimate the heading component of orientation, due to this angle being unaffected by the gravitational vector.

$$\theta_k = \text{atan2}(\alpha_{y_k}, \alpha_{z_k}) \tag{5.16}$$

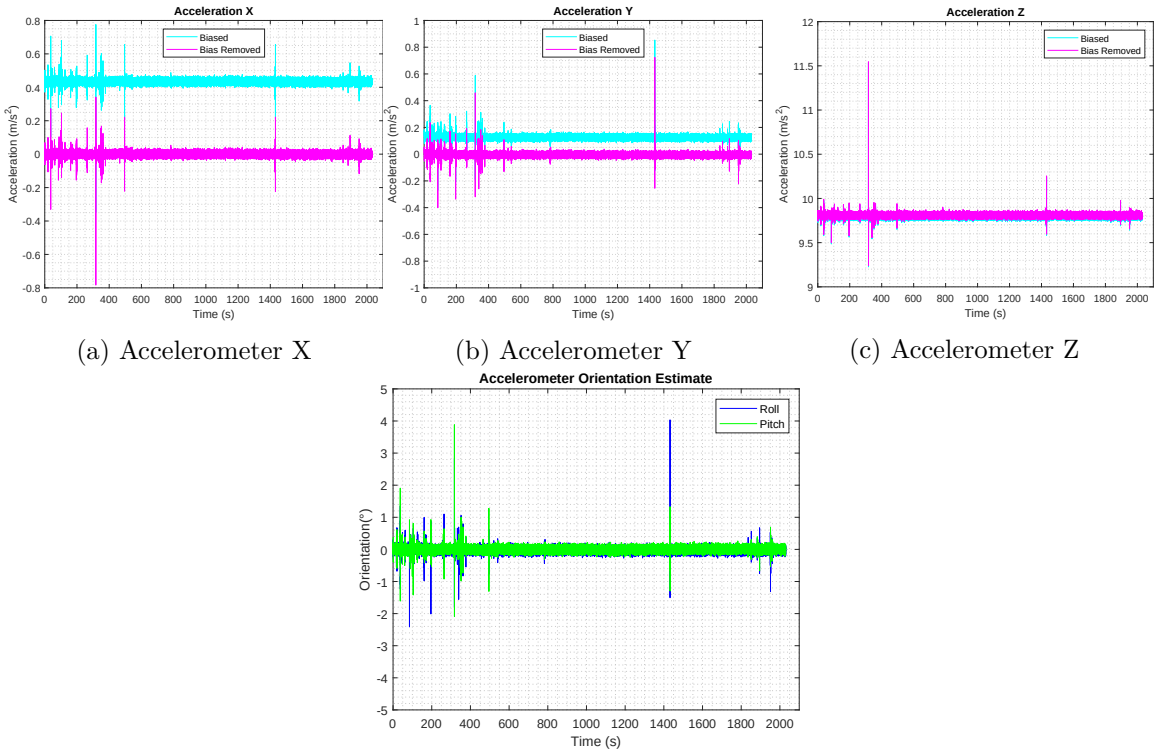
$$\phi_k = \text{atan2}(-\alpha_{x_k}, \sqrt{\alpha_{y_k}^2 + \alpha_{z_k}^2}) \tag{5.17}$$

Where;

- θ - Euler angle Roll (deg)
- ϕ - Euler angle Pitch (deg)
- k - Time instance (s)
- α_i - Acceleration $i \in [x,\ y,\ z]$ (m/s^2)

The static dataset for the three axis accelerometers acquired in Section 5.5 is now utilised to demonstrate the nature of the orientation output from the accelerometers. The statistics concerning the orientation output from the accelerometers are shown in Figure 5.27.

Within Figure 5.27, the sub-plots Figure 5.27a, Figure 5.27b and Figure 5.27c represent the acquired results for each axis accelerometer in Section 5.5. The representation of



(a) Accelerometer X (b) Accelerometer Y (c) Accelerometer Z

(d) Orientation from Accelerometers

Figure 5.27: Orientation from Accelerometers

these results is to provide a reference as to how all three components of acceleration were utilised in Equation (5.16) and Equation (5.17) for orientation calculations. The calculated output for orientation estimation from the accelerometers is shown in Figure 5.27 in degrees format. It is to be noted that the calculated orientation uses the bias removed datasets for each component of acceleration and such bias quantities were negated from each accelerometer observation. It is clear from analysing Figure 5.27 that the roll and pitch estimations lay around the 0° as expected in static conditions. However, the resulting roll and pitch experience perturbations similar to the nature of raw accelerometer output shown in Figure 5.27a, Figure 5.27b and Figure 5.27c. Additionally, the accelerometer orientation estimates do not deal well with outlier accelerometer readings, which can be observed by deviations in orientation.

$$\beta_\alpha = [0.434,\ 0.127,\ 0.014]^T \tag{5.18}$$

5.6.1.2 Gyroscope

A similar process can be applied to gyroscopic output in order to estimate orientation. Unlike accelerometers, gyroscopes naturally provide estimations concerning all three components of orientation. The method applied to achieve this is through the time integration of the angular velocity output provided by the gyroscopes on each axis to provide an incremental change in angle of orientation. The mathematical operation to achieve this is shown in Equation (5.19) and Equation (5.20) for the roll angle, Equation (5.21) and Equation (5.22) for the pitch angle and Equation (5.23) and Equation (5.24) for the heading angle. It is here that once the readings are extracted from the gyroscope in their native units of radians, they are converted into degrees to provide consistency with the accelerometer orientation output. It is to note that there are two denotations used to describe time in the following set of equations. These notations are t and k and are used to represent a time-step and global elapsed time respectively. These notations are consistently used throughout the remainder of Chapter 5.

$$\delta\theta_k = \omega_{x_k}\, t \tag{5.19}$$

$$\theta_k += \delta\theta_k \tag{5.20}$$

$$\delta\phi_k = \omega_{y_k}\, t \tag{5.21}$$

$$\phi_k += \delta\phi_k \tag{5.22}$$

$$\delta\psi_k = \omega_{z_k}\, t \tag{5.23}$$

$$\psi_k += \delta\psi_k \tag{5.24}$$

Where;

- $\omega_{[x,y,z]}$ - Angular velocity (deg/s)
- $\delta\theta$ - Change in Euler angle Roll (deg)
- $\dot{\phi}$ - Change in Euler angle Pitch (deg)
- t - Time increment (s)

The final calculated results for orientation from the static based gyroscopic characterisation test are shown in Figure 5.28.

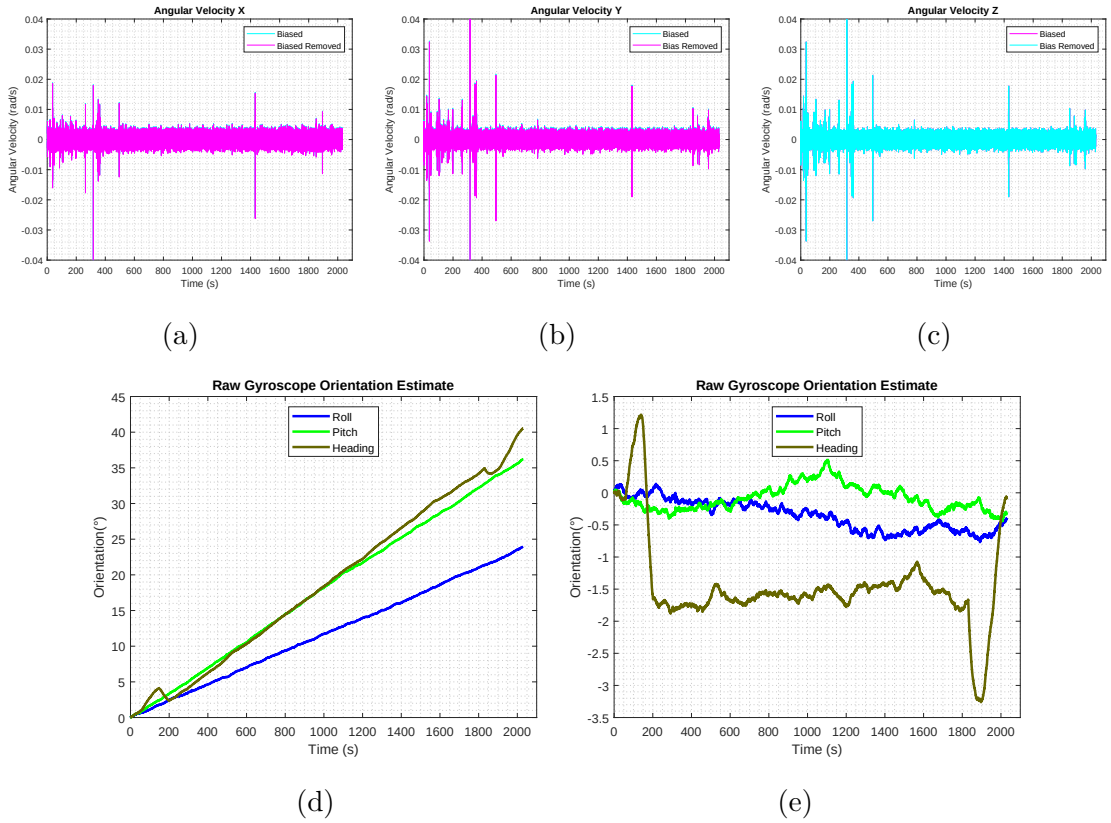


Figure 5.28: (a) Gyroscope output X (b) Gyroscope output Y (c) Gyroscope output Z (d) Gyroscope Orientation Estimate w/ Bias (e) Gyroscope Orientation Estimate w/o Bias

The sub-plots presented in Figure 5.28(a), Figure 5.28(b) and Figure 5.28(c) are the raw biased and unbiased outputs provided by the three axis gyroscope structure within the IMU on-board the mobile robot. They are represented for reference as these statistics are used as inputs into the calculation of orientation change for each axis shown in Equation (5.19), Equation (5.21) and Equation (5.23). The final calculated orientation estimations from raw gyroscopic output for the three axes are shown in Figure 5.28(d) and Figure 5.28(e). Although the gyroscopic orientation estimation output is of a smooth nature, it is evident that with the inclusion of gyroscopic biases, orientation estimates are seen to drift drastically over time. This is known as gyroscopic drift and is the result of the application of the time integration and the gyroscope not reading 0 when in static conditions. Even though the biases are relatively small and are almost unobservable as seen in Figure 5.28(a), Figure 5.28(b) and Figure 5.28(c), the drift effects can be significant over longer periods of time without the integration of other sensor systems such as accelerometers

which can aid in the detection of random walk drifting effects. All three estimations for each component of orientation are seen to experience gyroscopic drift over the course of the static characterisation test as expected. This is observed in Figure 5.28(d). Post removal of the gyroscopic bias, orientation estimates are not seen to experience as much of a significant drifting effect, with the roll and pitch angles lingering between $-0.5^\circ - 0.5^\circ$, this is evident in Figure 5.28(e). However, there are low-scale signs of small random walk occurrences. This may be due to the fact that gyroscopic biases are not consistent values, where these biases may change under temperature fluctuations, environment and when subjected to external mechanical loading.

5.6.1.3 Conclusions

In conclusion to Section 5.6.1, it was evident that orientation estimations from accelerometers and gyroscopes as stand-alone systems yield both beneficial aspects and also significant limitations. Estimations from the accelerometer proved to be accurate yet not very precise, whereas the gyroscope output demonstrated preciseness in terms of low levels of noise but suffered catastrophically to time integration drift when the gyroscopic biases were not handled correctly. Therefore, the next category of work located within Section 5.6.2 offers a solution in the form of a Kalman Filter (KF) sensor fusion process. However, the standard KF is not utilised but an effective variant of the KF is presented. Instead of modelling the state of the orientations, this KF variant analyses the error within the state estimations provided by the accelerometers and gyroscopes, it is known as the Error State Kalman Filter (ES-KF). This is executed to fuse both sets of readings in an attempt to complement the key assets that each sensor has to offer.

5.6.2 Error State Kalman Filter Formulation

The ES-KF or commonly referred to as the Indirect Kalman Filter (IKF) operates without the definition of the state vector or a motion model representing temporal state transitions, procedures which are implemented when deploying a standard and direct Kalman Filter approach. The method of filtering that the ES-KF deploys is known as feedback filtering, where the errors of the inertial system are estimated and used to correct the system's output [105]. The operation of the ES-KF seeks to estimate error statistics concerning the systems which are required to undergo the sensor fusion process [106]. Following on from

the orientation estimations acquired from the accelerometers and gyroscopes as stand-alone systems in Section 5.6.1, an ES-KF is designed to fuse accelerometer and gyroscopic output in an attempt obtain a more stable estimate for 3D orientation. The critical factor to understand concerning the ES-KF is the tracking and the estimation of the gravitation vector from both the accelerometers and gyroscopes. This enables the calculation of the error state residual. The deployment of an ES-KF for sensor fusion to estimate orientation is a fairly traditional approach utilised in the design of inertial navigation systems (INS) for land and aerial robotic systems [107]. Therefore, Section 5.6.2 presents the mathematical formulation of the ES-KF implemented in this thesis.

The error state vector used in the ES-KF is shown in Equation (5.25). It is to note that the inclusion of the ϵ notation represents error. The components of χ_ϵ comprise of the error state vector for orientation denoted as Φ_ϵ, β_ϵ denoting the vector for the gyroscopic zero rate offset error and α_ϵ used to represent the estimated linear acceleration error vector. Each component within χ_ϵ is represented by three sub-components for each dimension of 3D space, therefore making χ_ϵ a 9D error state vector.

$$\chi_\epsilon = [\Phi_\epsilon \ \beta_\epsilon \ \alpha_\epsilon]^T \tag{5.25}$$

The formulation of the ES-KF will be split into three main sections for simplicity. These sections are outlined below and are then presented in Section 5.6.2.1, Section 5.6.2.2 and Section 5.6.2.3.

- **A Priori Prediction** - Discusses and formulates the prediction stage of the ES-KF. Explains the importance of the gyroscope and accelerometers.
- **Error Residual** - Presents the error residual estimation and how it is calculated.
- **A Posteriori Update** - Formulates the correction step and how the error state is updated and overall 3D orientation is estimated.

5.6.2.1 A Priori Prediction

Within the prediction step of the ES-KF, it is important to note that the initial prediction of the orientation is estimated using the gyroscopic update. The orientation within an ES-KF is operated on in a quaternion format, which enables the ease of transformations

from world to sensor body frame. The model prediction is split into three sub-sections each classifying an important step during this stage of the ES-KF. These sub-sections are listed below;

- Predict Orientation from Gyroscope
- Estimate the Gravitational Vector from Predicted Orientation
- Estimate the Gravitational Vector from Accelerometer

The indirect variant of the KF algorithm sets the a priori estimate of the error state vector to zero at the start each iteration of the filter, as shown in Equation (5.26). This differs from the traditional KF mathematics where a system motion model is utilised to predict the state ahead. The underlying logic behind the contrasting mathematics is that the ES-KF estimates the error within each process, where the process is defined as each single filter iteration. Therefore, the error at the start of each iteration is equivalent to zero as measurements are yet to be received.

$$\widehat{\chi}_{\epsilon_k}^{-} = \begin{bmatrix} \widehat{\Psi}_{\epsilon_k}^{-} \\ \widehat{\beta}_{\epsilon_k}^{-} \\ \widehat{\alpha}_{\epsilon_k}^{-} \end{bmatrix} = 0 \tag{5.26}$$

With the a priori estimate of the error state vector being set to zero, this therefore results in the linear prediction matrix $\boldsymbol{A_k}$ natively utilised in the traditional format of a KF being set to zero. This is also equal to zero as presented in Equation (5.27). Therefore, the a priori error covariance is equal to the system process noise matrix Q_k. This is shown in Equation (5.28).

$$\boldsymbol{A_k} = 0 \tag{5.27}$$

$$\widehat{P}_k^{-} = \boldsymbol{A_k}\widehat{P}_{k-1}\boldsymbol{A_k}^T + Q_k = Q_k \tag{5.28}$$

The process noise variable Q_k, however, is determined by a function which takes into account the a posteriori error estimate covariance from the previous time step ($\widehat{P}_{k-1}$). Therefore, Q_k is updated in the previous time instance prior to the execution of the prediction stage of the ES-KF. This is represented by Equation (5.29).

$$Q_k = f(\widehat{P}_{k-1}, \nu, \eta) \tag{5.29}$$

Due to its size, the finalised matrix format of Q_k is shown in Appendix I(a) and its corresponding matrix components in Appendix I(b) due to its size.

Predict Orientation from Gyroscope

The prediction of the orientation using the gyroscope is achieved through the integration of the raw gyroscopic measurements, as shown in Equation (5.30). Equation (5.30) calculates an estimation of the small change in angle ($\delta\widehat{\Psi}$) generated by the gyroscopes through a time integration process using the IMU update time interval *t*. This integrates the previous a posteriori estimate of the 3x1 gyroscopic bias vector $\widehat{\beta}$ at time instance *k-1*.

$$\delta\widehat{\Psi}_k = (\omega_k - \widehat{\beta}_{k-1})\, t \tag{5.30}$$

$\delta\Psi_k$ is then converted to quaternion format through the native rotation vector to quaternion conversion in ZYX rotation order shown in Equation (5.31). Calculating orientation in quaternion format is a traditional method in the design of inertial navigation systems, where a quaternion is structured as $\mathbb{H} = w + xi + yj + zk;\ w, x, y, z \in \mathbb{R}$. This conversion provides a base format that the estimations from the accelerometers and gyroscopes can take to maintain consistency.

$$\delta q_k(\delta\widehat{\Psi}_k) = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix}_k = \begin{bmatrix} \cos(\frac{\psi}{2})\cos(\frac{\phi}{2})\cos(\frac{\theta}{2}) + \sin(\frac{\psi}{2})\sin(\frac{\phi}{2})\sin(\frac{\theta}{2}) \\ \cos(\frac{\psi}{2})\cos(\frac{\phi}{2})\sin(\frac{\theta}{2}) + \sin(\frac{\psi}{2})\sin(\frac{\phi}{2})\cos(\frac{\theta}{2}) \\ \sin(\frac{\psi}{2})\cos(\frac{\phi}{2})\sin(\frac{\theta}{2}) + \cos(\frac{\psi}{2})\sin(\frac{\phi}{2})\cos(\frac{\theta}{2}) \\ \sin(\frac{\psi}{2})\cos(\frac{\phi}{2})\cos(\frac{\theta}{2}) + \cos(\frac{\psi}{2})\sin(\frac{\phi}{2})\sin(\frac{\theta}{2}) \end{bmatrix}_k \tag{5.31}$$

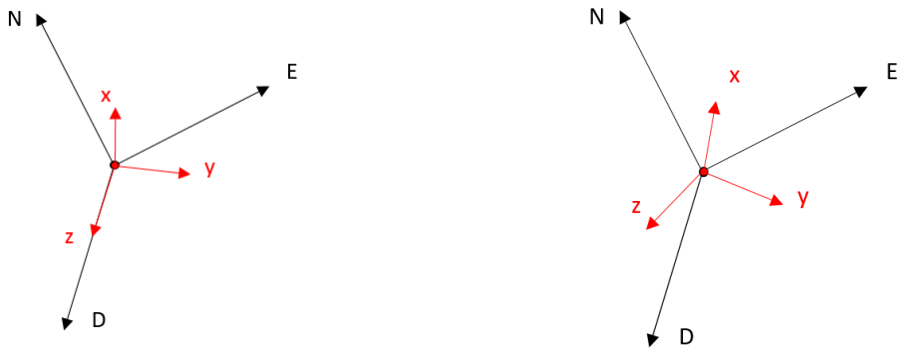
Once the change in orientation has been converted into a quaternion format denoted as δq_k, the orientation is then predicted prior to the calculation of the error residual. This is calculated using Equation (5.32) which executes a quaternion multiplication to rotate the previous a posteriori orientation estimate $\widehat{q}_{k-1}$ by δq_k.

$$\widehat{q}_{\bar{k}} = \widehat{q}_{k-1} \times \delta q_k(\delta\widehat{\Psi}_k) \tag{5.32}$$

The value $\widehat{q}_{\bar{k}}$ denotes the a priori orientation estimate at time k.

Estimate the Gravitational Vector from Gyroscope Predicted Orientation

The error residual is calculated based on the difference between the estimation of the gravitation vector from the gyroscope and the accelerometer. Considering the gyroscope, an estimation of the gravity vector is to be extracted from the predicted orientation previously calculated in Equation (5.32). This process sees the calculation of the gravity vector from an orientation quaternion and is achieved through the inclusion of the fixed gravitational vector in the world frame with notation North East Down (NED). The difference between the two important reference frames is displayed in Figure 5.29 with the black and red coordinate axes denoting the world and sensor body frame respectively. When the sensor body frame is perfectly aligned with that of the world frame as portrayed in Figure 5.29a, the gravitational vector in the sensor body frame is equivalent to that of the world frame which is denoted as $g^{\boldsymbol{W}}$, this is shown in Equation (5.33), where $g_k^{\boldsymbol{B}}$ represents the gravitational vector at time k in the sensor body frame (B) in units of gravitational acceleration (G), where the metric 1G = 9.81m/s^2 was used, it is to note that this figure can vary depending upon the location on Earth. However, when perfect alignment does not occur and there is an offset between the two frames, a scenario similar to that shown in Figure 5.29b arises and in this case, the gravitational vector in the sensor body frame will be represented as some quantity of acceleration on three axes as shown in Equation (5.34).



(a) (b)

Figure 5.29: World and sensor body frames (a) Sensor & World frame alignment (b) Sensor Frame offset

$$g_k^{\boldsymbol{B}} = g^{\boldsymbol{W}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{5.33}$$

$$g_k^{\boldsymbol{B}} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{5.34}$$

The reason for the presentation of Figure 5.29, Equation (5.33) and Equation (5.34) stems back to estimating gravity from the predicted orientation provided by Equation (5.32). The output of Equation (5.32) represents the current predicted rotational misalignment between the sensor body and world frames in the traditional orientation quaternion format. However, the gravitational estimation from quaternion can be simplified through representing the quaternion in rotational matrix format. The estimated gravitation vector from the predicted orientation calculated in Equation (5.32) denoted as ${}^{\boldsymbol{G}}g^{\boldsymbol{B}}$ is shown in Equation (5.35), where the superscript G indicates that it is an estimate acquired from the gyroscope as opposed to the accelerometer.

$${}^{\boldsymbol{G}}g_k^{\boldsymbol{B}} = \widehat{\boldsymbol{R}}_{\bar{k}}^{\boldsymbol{G}}\, g \tag{5.35}$$

Where $\widehat{\boldsymbol{R}}_{\bar{k}}^{\boldsymbol{G}}$ is the estimated a priori rotation matrix conversion of $\widehat{q_{\bar{k}}}$ calculated through Equation (5.36) and w, x, y and z are the components of $\widehat{q_{\bar{k}}}$.

$$\widehat{\boldsymbol{R}}_{\bar{k}}^{\boldsymbol{G}} = \begin{bmatrix} 2w^2 - 1 + 2x^2 & 2xy - 2wz & 2xz + 2wy \\ 2xy - 2wz & 2w^2 - 1 + 2y^2 & 2yz + 2wx \\ 2xz - 2wy & 2yz + 2wx & 2w^2 - 1 + 2z^2 \end{bmatrix} \tag{5.36}$$

Estimate the Gravitational Vector from Accelerometer

The estimation of the gravitational vector from the accelerometer is a much more elementary process as opposed to that of the gyroscope. The estimated gravitational vector from the accelerometer is as simple as acquiring the current output from the accelerometer (α_{y_k}) and negating the 3x1 accelerometer bias (β_α). However, the estimation of the linear acceleration (α_k) is also taken into account. This is because if the system on which the IMU is mounted is dynamically operating, then any linear accelerations due to translational motion can affect the accelerometer's gravitational vector estimate. The underlying reason behind this perturbation is that the accelerometer perceives this linear acceleration as a

result from a change in orientation, therefore, it needs to be considered. The gravitational vector estimate from the accelerometer is calculated through Equation (5.37), where A denotes that this value is acquired from the accelerometer.

$$^{\boldsymbol{A}}g_k^{\boldsymbol{B}} = (\alpha_{yk}^{\boldsymbol{B}} - \beta_\alpha) - \alpha_k^{\boldsymbol{B}} \tag{5.37}$$

5.6.2.2 Error Residual

The error residual denoted as Y is the difference between the gravitational vector estimations produced from both the gyroscope and the accelerometers presented in the previous two sub-sections. This difference calculation is presented in Equation (5.38).

$$Y_{\epsilon_k} = {}^{\boldsymbol{G}}g_k^{\boldsymbol{B}} - {}^{\boldsymbol{A}}g_k^{\boldsymbol{B}} \tag{5.38}$$

5.6.2.3 A Posteriori Update

The a posteriori update within the ES-KF is mainly where generic Kalman implementations are introduced within the recursive process. The prior estimation of the orientation is updated through the calculation of the innovation covariance (S_k), Kalman Gain (K_k) and the observation model (H_k), where k denotes a specific time instance. Therefore, Section 5.6.2.3 is split into 4 separate sub-sections which are outlined below.

- Calculate Kalman Equations
- Estimate A Posteriori Error State Vector
- Calculate A Posteriori States
- Update A Posteriori Error Covariance Matrix

Calculate Kalman Equations

This sub-section within the update stage of the ES-KF handles the calculation of the Kalman variables required in order for the filter to operate. These equations are consistent with the traditional KF formulation. Equation (5.39) presents the calculation of the innovation covariance required prior to the calculation of the Kalman Gain shown in Equation (5.40). The Kalman Gain is the weighting factor used to update the a priori estimate of the error state vector.

$$S_k = R_k + H_k \widehat{P}_k^- H_k^T \tag{5.39}$$

$$K_k = \widehat{P}_k^- H_k^T S_k^{-1} \tag{5.40}$$

For the mathematics presented in Equation (5.39) and Equation (5.40), the matrices H_k and R_k are required. R_k represents the observation model and its corresponding noise covariance matrix and is shown in Equation (5.41).

$$R_k = \begin{bmatrix} \nu + \eta & 0 & 0 \\ 0 & \nu + \eta & 0 \\ 0 & 0 & \nu + \eta \end{bmatrix} \tag{5.41}$$

Notations ν and η within Equation (5.41) represent the accelerometer and gyroscopic observation noise variances, such values were quantified during the IMU characterisation procedures presented in Section 5.5. Therefore, observation error standard deviations (σ) for accelerometer (α) and gyroscope angular rates (ω) that were previously defined in Equation (5.14) and Equation (5.15) are redefined in Equation (5.42) and Equation (5.43) for reference.

$$\sigma_{\alpha_{x,y,z}} = [0.0128,\ 0.0135,\ 0.0191]^T \tag{5.42}$$

$$\sigma_{\omega_{x,y,z}} = [0.0014,\ 0.00145,\ 0.00195]^T \tag{5.43}$$

Therefore, ν and η are calculated through averaging $\sigma_{\alpha_{x,y,z}}$ and $\sigma_{\omega_{x,y,z}}$. Final values are shown in Equation (5.44) and Equation (5.45). The reason for squaring the output from the average calculation is that the averaged values represented standard deviation of error and the required metric to be set within the covariance matrix R_k is the error variance.

$$\nu = \frac{0.0128 + 0.0135 + 0.0191}{3} = 0.01516^2 \text{ m/s}^2 \tag{5.44}$$

$$\eta = \frac{0.0014 + 0.00145 + 0.00195}{3} = 0.0016^2 \ rad/s \tag{5.45}$$

The observation model H_k is shown in Equation (5.46). The nomenclature (${}^{\boldsymbol{G}}g_k^{\boldsymbol{B}}\times$) denotes the skewed symmetric matrix of ${}^{\boldsymbol{G}}g_k^{\boldsymbol{B}}$ represented as that shown in Equation (5.47).

$$H_k = \begin{bmatrix} ({}^{\boldsymbol{G}}g_k^{\boldsymbol{B}}\times)_{3\times3} & t({}^{\boldsymbol{G}}g_k^{\boldsymbol{B}}\times)_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{5.46}$$

$${}^{\boldsymbol{G}}g_k^{\boldsymbol{B}}\times = \begin{bmatrix} 0 & {}^{\boldsymbol{G}}g_z^{\boldsymbol{B}} & -{}^{\boldsymbol{G}}g_y^{\boldsymbol{B}} \\ -{}^{\boldsymbol{G}}g_z^{\boldsymbol{B}} & 0 & {}^{\boldsymbol{G}}g_x^{\boldsymbol{B}} \\ {}^{\boldsymbol{G}}g_y^{\boldsymbol{B}} & -{}^{\boldsymbol{G}}g_x^{\boldsymbol{B}} & 0 \end{bmatrix} \tag{5.47}$$

Estimate A Posteriori Error State Vector

The estimation of the a posteriori error state vector is achieved through the combination of the Kalman Gain and the error residual calculated in Equation (5.40) and Equation (5.38). This updating process is shown in Equation (5.48).

$$\widehat{\chi}_{\epsilon_k} = \begin{bmatrix} \widehat{\Psi}_{\epsilon_k} \\ \widehat{\beta}_{\epsilon_k} \\ \widehat{\alpha}_{\epsilon_k} \end{bmatrix} = K_k Y_{\epsilon_k} \tag{5.48}$$

Calculate A Posteriori States

The states set to be outputs of the ES-KF which are the orientation of the IMU as well as estimations of the gyroscope zero offset bias and linear accelerations are then updated using the result acquired from Equation (5.48). For the orientation, this is achieved by negating the a posteriori error in rotation $\widehat{\Psi}_{\epsilon_k}$ from the current a priori orientation estimate $\widehat{q_{\bar{k}}}$. This once again transforms $\widehat{\Psi}_{\epsilon_k}$ into quaternion format through Equation (5.31). This orientation update is achieved through Equation (5.49).

$$\widehat{q_k} = \widehat{q_{\bar{k}}} \times \delta q_k(-\widehat{\Psi}_{\epsilon_k}) \tag{5.49}$$

The similar process is carried out for the estimation of the gyroscopic zero rate offset and the linear acceleration, where the corresponding error state components are negated in order to execute the update process. This is shown in Equation (5.50).

$$\begin{bmatrix} \widehat{\beta}_k \\ \widehat{\alpha}_k \end{bmatrix} = \begin{bmatrix} \widehat{\beta}_k^- \\ \widehat{\alpha}_k^- \end{bmatrix} - \begin{bmatrix} \widehat{\beta}_{\epsilon_k} \\ \widehat{\alpha}_{\epsilon_k} \end{bmatrix} \tag{5.50}$$

Update A Posteriori Error Covariance Matrix

The final value to be updated is the error covariance matrix. Equation (5.51) presents

the calculation of the a posteriori error covariance matrix $\widehat{P}_k$ prior to beginning the next iteration of the filter.

$$\widehat{P}_k = (I - K_k H_k \widehat{P}_k^-) \tag{5.51}$$

5.6.3 Error State Kalman Filter Results

Section 5.6.3 presents the initial results gathered from the implementation of the ES-KF as well as acquiring the error statistics for covariance matrix generation. It demonstrates how the ES-KF reduces noise in estimations of orientation from the IMU based on the static dataset acquired during the IMU characterisation process. The results from the ES-KF are then assessed to generate variance metrics prior to its inclusion within the 6DOF-EKF. Further results from the ES-KF and how it stabilises the final orientation estimates when fused with the LiDAR are expanded upon in Section 5.7.

5.6.3.1 ES-KF Orientation Estimation

Initial results acquired from the ES-KF are shown in Figure 5.30 based on the static dataset acquired over a time period of approximately 2,000s. The previously presented orientation estimates from the accelerometers and gyroscopes as stand-alone systems are re-presented in Figure 5.30a and Figure 5.30b for comparison.

It is evident that when observing the output from the ES-KF shown in Figure 5.30c that the resulting orientation estimates significantly reduce the noise from the accelerometers and simultaneously reduce the effect of gyroscopic drift. It can be stated that the ES-KF combines the precise but inaccurate estimates from the gyroscope with the noisy yet accurate estimates from the accelerometer, resulting in estimates which are closer to the true value of 0° in roll and pitch for a static based experiment. However, the output for the heading component of motion differs. Due to the accelerometer only being able to provide estimations concerning the roll and pitch angles, this therefore results in the scenario where there is not a secondary heading measurement present to accompany that provided by the gyroscope.

Additionally, an example of the consequences when mishandling proprioceptive sensor biases is presented in Figure 5.31. It demonstrates that with the inclusion of sensor bias, the ES-KF assumes that the sensor body is in a maintained rotated position for both roll,

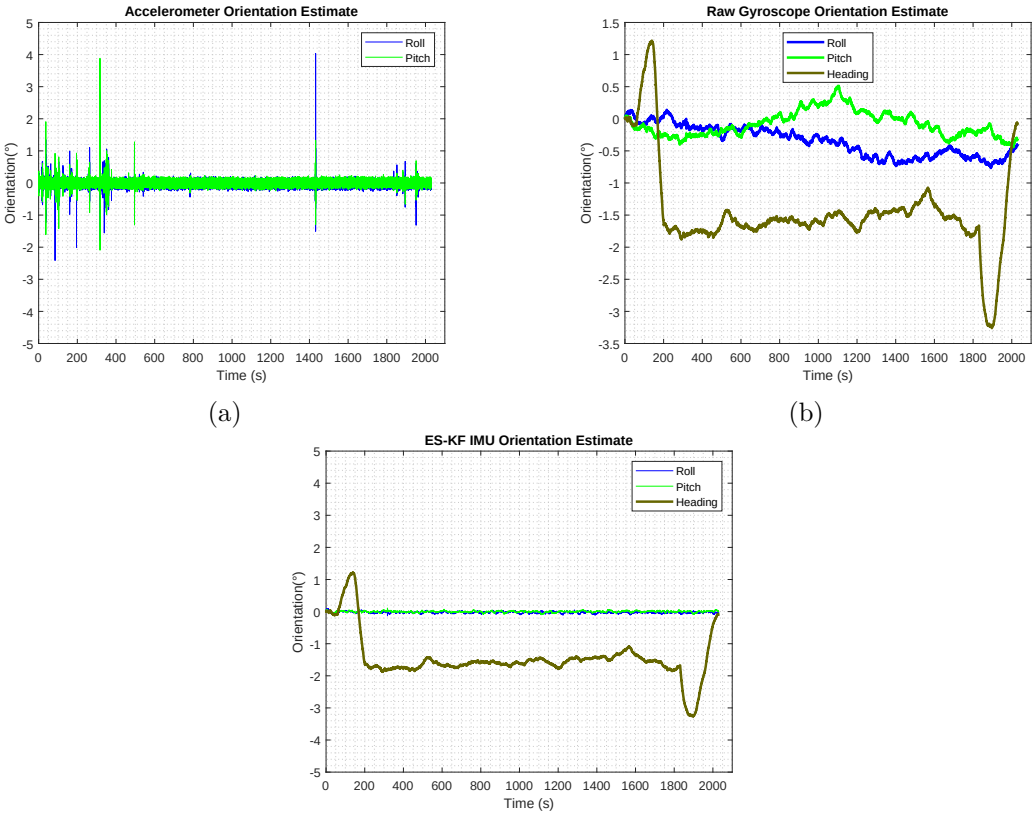


Figure 5.30: Orientation Estimates (a) Accelerometer (b) Gyroscope (c) ES-KF

and pitch angles. This can cause major problems when the algorithm is set to operate as orientation estimations will be completely offset.

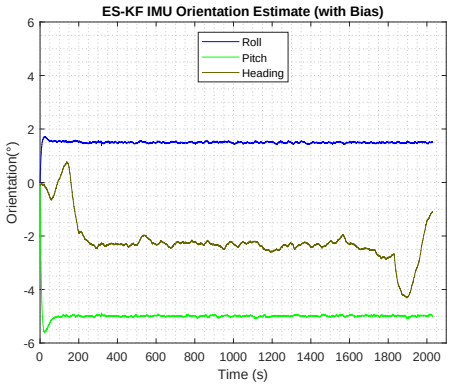
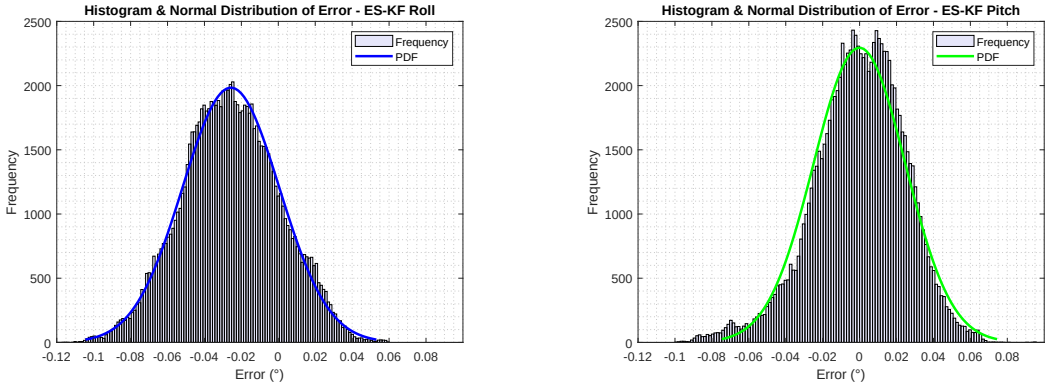


Figure 5.31: ES-KF Orientation Estimate with Accelerometer Bias

5.6.3.2 ES-KF Error Quantification

The final aspect in the formulation and presentation of the initial ES-KF results is quantification of the ES-KF final error metric. The process of quantifying the error metric for each designed algorithm of output from a motion estimation technique may have seemed

a recursive process throughout this Chapter. However, the correct characterisation of these error metrics enables the design of a solid and effective 6DOF-EKF as traditional techniques are seen to change random variables until the filter outputs satisfactory results. Therefore, Section 5.6.3.2 presents the mean and standard deviations of error for the ES-KF results presented in Section 5.6.3.1. The nature of the estimation error acquired from the ES-KF is shown in Figure 5.32, specifically Figure 5.32a and Figure 5.32b for the roll and pitch angle estimations respectively. Results show that the final error metrics are on a sub-degree scale when subjected to static conditions. However, the roll angle estimations are seen to possess a mean error which is taken into account as a bias metric. This can be seen in the slight shift away from zero of the histogram plot shown in Figure 5.32a by 0.025° as opposed to the pitch angle which produced a zero mean distribution shown in Figure 5.32b. Once again, both angle estimation errors produced a distribution of a Gaussian nature, this can be observed by the shape of the histogram frequencies Figure 5.32a and Figure 5.32b. Finally, final error statistics for the roll and pitch angle estimations from the ES-KF can be seen in Table 5.7.



(a) (b)

Figure 5.32: ES-KF Error Statistics (a) Roll (b) Pitch

Table 5.7: ES-KF Orientation Error Statistics

Component	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$
θ (°)	-0.025	2.75×10^{-2}
ϕ (°)	-8.687×10^{-5}	2.74×10^{-2}

5.6.4 Conclusion

In conclusion, Section 5.6.2 introduced the sensor fusion process of accelerometers and gyroscopes through what is known as the ES-KF. Section 5.6.2 also presented the underlying function of the ES-KF, its formulation, the initial results, its error variance and how this algorithm can be an effective component within an orientation estimation system. Initial results proved that the ES-KF significantly produces improved orientation estimates as opposed to accelerometers and gyroscopes as stand-alone systems. This ES-KF sensor fusion process adds in an extra source of information to aid in the estimation of the orientation components of motion within the 6DOF-EKF. The values calculated within Section 5.6.3 concerning the error statistics for the ES-KF are outlined below, where σ_θ, σ_ϕ and σ_ψ represent the resulting standard deviations of error for the roll, pitch and heading orientation estimations respectively. These values are used in the design of the 6DOF-EKF.

- $\sigma_\theta = 2.75 \times 10^{-2^\circ}$
- $\sigma_\phi = 2.74 \times 10^{-2^\circ}$
- $\sigma_\psi = 7.16 \times 10^{-1^\circ}$

The following section now presents the formulation of the 6DOF-EKF, which estimates six components of motion for the mobile robotic system. The ES-KF is integrated into the 6DOF-EKF by providing observations estimations for roll and pitch orientation. This is explained throughout Section 5.7.

5.7 Six Degree-of-Freedom EKF Formulation

Generating kinematic models for the platforms designed for this thesis enabled the identification on how the platforms' states would transition over time. After introducing the theory on the behaviour of the ICRs and wheel slippage of skid-steered robots in Chapter 3, the decision was taken to model the foundations of the platforms' kinematics based on the method introduced in [35]. Additionally, the method undertaken does not include wheel slippage and ICR estimations as possible wheel slippages occurrence was factored in as a component of the process noise within the formulation of the motion model that will now be presented. The models generated in this section were used as the foundations for the

design of the state estimation algorithms used to approximate the pose of the system as it navigates throughout an experimental environment. The components of the state of the platform χ at time k is shown in Equation (5.52), where the modelled states of the platform include its 3D Cartesian position and its 3D angular position, all making up the 6DOF pose of a mobile system.

$$\chi_k = [x_k \; y_k \; z_k \; \theta_k \; \phi_k \; \psi_k]^T \tag{5.52}$$

Where;

- x_k - Position in x (m)
- y_k - Position in y (m)
- z_k - Position in z (m)
- θ_k - Roll (deg)
- ϕ_k - Pitch (deg)
- ψ_k - Heading (deg)

When modelling the platform's kinematics, a system control input is utilised to aid in the estimation of the platform's orientation transition over time. The control input vector u_k, shown in Equation (5.53) includes the acquired output from the gyroscopes. For each observation from the IMU, the ES-KF estimates the orientation and feeds it into the 6DOF-EKF as a control input. Along with the rate of attitude change from the gyroscopes, u_k is also comprised of the linear displacement of each wheel, acquired from the angular velocity output from the quadrature encoders. The calculation executed to extract encoder linear displacement estimations from raw angular displacement outputs is an angular motion to linear motion conversion. This is carried out within the motion model presented in Section 5.7.1.

$$u_k = [\gamma_{L_k} \gamma_{R_k} \; \omega_k]^T \tag{5.53}$$

Where;

$$\omega_k = [\dot{\theta}_k \ \dot{\phi}_k \ \dot{\psi}_k]^T \tag{5.54}$$

Furthermore;

- γ_{L_k} - Angular displacement of the left wheel (deg)
- γ_{R_k} - Angular displacement of the right wheel (deg)
- ω_k - 3 axis gyroscope (deg/s)

5.7.1 Motion Model

As presented as a main point throughout this thesis, the process of system identification is a vital aspect when conducting studies of mobile robotic systems. Identical to that presented in Chapter 4, the state space motion model of the unmanned platform represents the transitions of the states within χ through the formulation of the system's kinematic behaviour. The information concerning orientation updates acquired from the gyroscopes is then integrated into the motion model in order to estimate the temporal transition of angles within χ. The generalised formulation of the motion model is shown in Equation (5.55), where $\widehat{\chi}_k^-$ and $\widehat{\chi}_{k-1}$ are the future predicted and previous estimated states.

$$\widehat{\chi}_k^- = f(\widehat{\chi}_{k-1}, u_k) \tag{5.55}$$

Therefore, utilising the generalised format shown in Equation (5.55), the integration of the encoder measurements and the output from the gyroscopes enable the formulation of the state transition function shown in Equation (5.56).

$$\widehat{\chi}_k^- = \widehat{\chi}_{k-1} + \begin{bmatrix} \delta D_k \ \cos\left(\psi_k + \frac{\delta\psi_k\, t}{2}\right) \\ \delta D_k \ \sin\left(\psi_k + \frac{\delta\psi_k\, t}{2}\right) \\ \delta D_k \ \sin(\phi) \\ \dot{\theta}_k t \\ \dot{\phi}_k t \\ \dot{\psi}_k t \end{bmatrix} \tag{5.56}$$

An important factor to consider within the motion model is the transition of the first and second elements of χ shown in Equation (5.56), where the inclusion of δD_k is seen to

reside. δD_k represents the change in forward-facing displacement of the unmanned system. The calculation and formulation of δD_k is related to the fixed forward facing direction of the system. Taking into account the sensing systems used as proprioceptive input, Figure 5.33 shows the set-up of each proprioceptive sensor in relation to their operating reference frame.

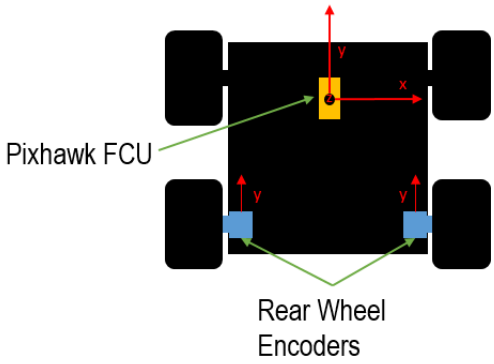


Figure 5.33: Proprioceptive Sensor Orientations

It was ensured that all the proprioceptive sensors were mounted with the same orientation to prevent confusion and misaligned reference frames between each sensor. It is to be noted that the motion model makes assumptions that the unmanned system is not to translate laterally or possess the capabilities to directly change its altitude, which systems such as UAVs possess. Therefore, considering the robot's lateral constraints due to its unidirectionality characteristics, observations from the quadrature encoders concerning forward-facing angular displacement of the unmanned system's wheels are used to calculate δD_k within the prediction stage of the 6DOF-EKF. The calculation of δD_k is shown in Equation (5.57).

$$\delta D_k = \frac{(\gamma_{L_k} r + \gamma_{R_k} r)}{2} \tag{5.57}$$

Where;

- r - Wheel Radius = 0.125m

The calculation of the linear displacement δD_k is also used in the prediction of the change in altitude of the unmanned system. Considering the design of the system in relation to its ground-based and unidirectionality characteristics, constraints can be set which relate δD_k to the current estimate of the system's pitch angle (ϕ). This can be seen in Equation (5.56) during the calculation of the altitude transition. An overview of the process can be seen in

Figure 5.34.

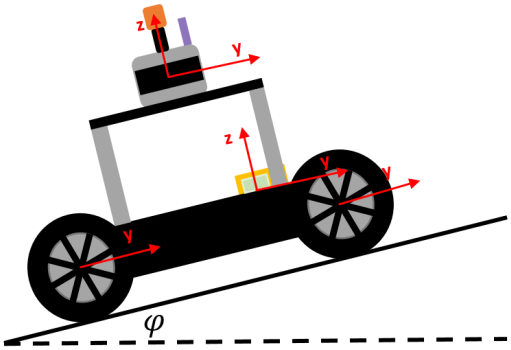


Figure 5.34: Structure of altitude change estimation

The structure of the motion model shown in Equation (5.56) uses factors concerning the heading within trigonometric equations to calculate the change in Cartesian position. This can be seen to be of a non-linear nature, therefore justifying the necessity for a non-linear sensor fusion and state estimation algorithm such as the EKF. The EKF employs a first-order Taylor expansion in the form of Jacobian matrices to linearly approximate the non-linear function. Within the motion there are two Jacobian matrices required to be calculated. The first is the partial derivative of each component of the motion model with respect to each component within the state vector χ . This is shown in a generalised form in Equation (5.58) at time instance k.

$$G_{x_k} = \frac{\partial f}{\partial \chi}(\widehat{\chi}_k, u_k) \tag{5.58}$$

Due to size characteristics of the final formulation of G_{x_k}, the matrix is simplified with the addition of place-holder representations for each index. The final formulation for G_{x_k} is shown in Equation (5.59).

$$G_{x_k} = \begin{bmatrix} \mathbf{0}_{3\times3} & \boldsymbol{G}_{a3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{5.59}$$

The place-holder G_a located within G_{x_k} is shown in Equation (5.60), where it is further simplified by three sub-place-holders to reduce presentation complexity. These sub-place-holders are shown in Equation (5.61), Equation (5.62) and Equation (5.63).

$$\boldsymbol{G}_a = \begin{bmatrix} 0 & 0 & \boldsymbol{G}_{a_2} \\ 0 & 0 & \boldsymbol{G}_{a_3} \\ \boldsymbol{G}_{a_1} & 0 & 0 \end{bmatrix} \tag{5.60}$$

$$G_{a_1} = \delta D_k \ \cos(\phi + \dot{\phi} t) \tag{5.61}$$

$$G_{a_2} = -\frac{\delta D_k \ \sin\left(\psi + \frac{\dot{\psi} t}{2}\right)}{2} \tag{5.62}$$

$$G_{a_3} = \frac{\delta D_k \ \cos\left(\psi + \frac{\dot{\psi} t}{2}\right)}{2} \tag{5.63}$$

The second Jacobian matrix within the motion model is the control odometry input Jacobian. As the temporal transitions of the components within the state vector are calculated from the control odometry input sources, partial derivatives of each component of the state transition function are executed with respect to the components of the input odometry vector u_k. The generalised form for the calculation of G_{u_k} is shown in Equation (5.64).

$$G_{u_k} = \frac{\partial f}{\partial u}(\widehat{\chi}_k, u_k) \tag{5.64}$$

However, the size of the generated matrix G_{u_k} also proved to be difficult to display and therefore followed the identical representation procedure of G_{x_k} through the use of place-holders for each matrix index. This representation for G_{u_k} is shown in Equation (5.65).

$$G_{u_k} = \begin{bmatrix} \boldsymbol{G}_{b3\times3} & \boldsymbol{G}_{d3\times2} \\ \boldsymbol{G}_{c3\times3} & \boldsymbol{G}_{e3\times2} \end{bmatrix} \tag{5.65}$$

The place-holder letter labelling system followed on from that of G_{x_k}. Prior to the description of each place-holder index, it is to be noted that the matrix index represented with $I_{3\times3}$ denotes an identity matrix and this is stated for reference purposes only. The first place-holder for G_{u_k} is represented by G_b as shown in Equation (5.66). The place-holder G_b also comprises of sub-place-holders demonstrating how the system variables are integrated. The sub-place-holders for G_b are shown in Equations (5.67) to (5.72).

$$\boldsymbol{G}_b = \begin{bmatrix} G_{b_1} & G_{b_2} & 0 \\ G_{b_3} & G_{b_4} & 0 \\ G_{b_5} & G_{b_6} & 0 \end{bmatrix} \tag{5.66}$$

$$G_{b_1} = \frac{\cos\left(\psi + \frac{\dot{\psi}\mathrm{t}}{2}\right)}{2} \tag{5.67}$$

$$G_{b_2} = G_{b_1} \tag{5.68}$$

$$G_{b_3} = \frac{\sin\left(\psi + \frac{\dot{\psi}\mathrm{t}}{2}\right)}{2} \tag{5.69}$$

$$G_{b_4} = G_{b_3} \tag{5.70}$$

$$G_{b_5} = \frac{\sin(\phi + \dot{\phi}\mathrm{t})}{2} \tag{5.71}$$

$$G_{b_6} = G_{b_5} \tag{5.72}$$

This process is also used to represent the final place-holders within G_{u_k} denoted as $\boldsymbol{G}_{c3\times3}$ shown in Equation (5.73).

$$\boldsymbol{G}_c = \begin{bmatrix} 0 & 0 & t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.73}$$

Placeholder $\boldsymbol{G}_{d3\times3}$ is represented by Equation (5.74).

$$\boldsymbol{G}_d = \begin{bmatrix} 0 & G_{d_1} \\ 0 & G_{d_2} \\ G_{d_3} & 0 \end{bmatrix} \tag{5.74}$$

The corresponding sub-place-holders within $\boldsymbol{G}_d$ are shown in Equations (5.75) to (5.77).

$$G_{d_1} = -\frac{\delta D_k \ \sin\left(\psi + \frac{\dot{\psi}t}{2}\right)}{2} \tag{5.75}$$

$$G_{d_2} = \frac{\delta D_k \ \cos\left(\psi + \frac{\dot{\psi}t}{2}\right)}{2} \tag{5.76}$$

$$G_{d_3} = \delta D_k \ \cos(\phi + \dot{\phi}t) \tag{5.77}$$

Finally, the remaining placeholder within G_{x_k} denoted as $\boldsymbol{G}_e$ is shown in Equation (5.78).

$$\boldsymbol{G}_e = \begin{bmatrix} 0 & 0 \\ t & 0 \\ 0 & t \end{bmatrix} \tag{5.78}$$

As a result of the on-board sensor error characterisation procedures carried out earlier in this chapter, the noise variance or uncertainty metric from the sensors can be integrated into the 6DOF-EKF. For the motion model, an integral variable that is required to be formulated is the process noise matrix Q. Q accounts for motion model uncertainties and is used to predict the state covariance matrix P. Within the motion model, the predicted states are calculated using the control input, which are encoder measurements as well as gyroscopic output from the IMU. Therefore, the motion model is only as good as the control input, therefore formulation of Q depends upon the variances from the control input. The covariance matrix for the control input source is shown in Equation (5.79), where σ represents the stand deviation of error for each component of the control input. The calculated values for σ of each component were achieved when characterising the gyroscopic output in Section 5.6.1.2.

$$U = \begin{bmatrix} \sigma_\gamma^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\gamma^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\dot{\theta}}^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{\phi}}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\dot{\psi}}^2 \end{bmatrix} \tag{5.79}$$

- $\sigma_\gamma = 5 \times 10^{-1}$ (deg/s)
- $\sigma_\theta = 8.01 \times 10^{-2}$ (deg)

- $\sigma_{\phi} = 8.31 \times 10^{-2}$ (deg)
- $\sigma_{\psi} = 6.87 \times 10^{-2}$ (deg)

0.0801 Therefore, the process noise covariance matrix Q is formulated through Equation (5.80).

$$Q = G_{u_k} \, U \, G_{u_k}^{T} \tag{5.80}$$

Due to the size of the matrix, the resulting output for Q can be found in Appendix II.

5.7.2 Observation Model

The observation model is concerning the source of observations that are provided to correct the predicted state estimate acquired from the motion model. For the case of the 6DOF-EKF, the sensory sources which provide the observations is a combination of the LiDAR system and the output from the ES-KF. The point cloud data acquisition from the LiDAR is integrated into the LiDAR odometry algorithm which outputs estimations concerning the displacement for 2D pose and the ES-KF output's roll and pitch angles. The observation for the Z position or elevation change is calculated using a fusion of LiDAR and ES-KF. The observation vector acquired at time instance k is shown in Equation (5.81). The observation concerning elevation displacement on the Z axis is calculated using the change in forward facing translational motion (Y axis) acquired from the LiDAR with the pitch angle estimation produced from the ES-KF. This is also shown below in Equation (5.81).

$$Z_k = [x \; y \; z \; \theta \; \phi \; \psi]^T \tag{5.81}$$

Where;

$$z = \dot{y} \; t \; \sin(\phi) \tag{5.82}$$

- x - X position (m)
- y - Y position (forward facing) (m)
- $\dot{y}$ - Rate of change Y position (m/s)
- z - Z position (m)

- θ - Roll orientation X axis (deg)
- ϕ - Pitch orientation Y axis (deg)
- ψ - Heading orientation Z axis (deg)
- t - Time step (s)

It is to be noted that the orientation of the LiDAR system was mounted and within the same operating reference frame as the IMU and the quadrature wheel encoders. This enabled constant and effective tracking and the prevention of confusion between both operating co-ordinate systems of each sensor, enabling the output of the ES-KF and wheel encoders to be easily integrated with the LiDAR system. The orientation reference frame of the LiDAR is shown in Figure 5.35, this can be referred to Figure 5.33 to observe consistent sensor co-ordinate frame orientations.

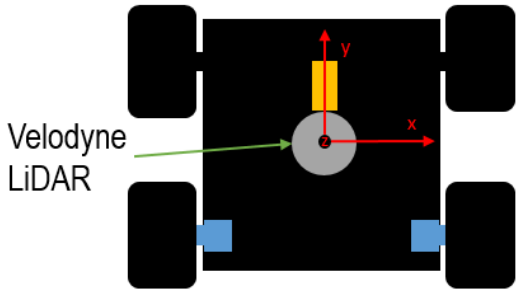


Figure 5.35: Observation sensor frame LiDAR

The sensor fusion aspect of any EKF relies of the calculation of a residual metric. This residual metric is the difference between the motion model state prediction and what the observations from the sensors measure. Therefore, there needs to be a formulation of a function which relates the components of the state vector to the observation vector. This function is known as the observation function and is shown in its generalised formulation in Equation (5.83), where $\widehat{Z_k}$ represents a set of predicted observations.

$$\widehat{Z_k} = h(\widehat{\chi_k^-}) \tag{5.83}$$

The observation function showing the relationship between the components within the state vector and the components within the observation vector is shown in Equation (5.84).

This function therefore enables the calculation of a residual in preparation for the state estimate correction that takes place within the observation model.

$$h(\widehat{\chi}_k^-) = \boldsymbol{I_{6\times 6}} \tag{5.84}$$

Although the observation function $h(\widehat{\chi}_k^-)$ is of a linear nature in the form of an identity matrix, there is still a requirement to calculate a Jacobian within the observation model. This is due to the non-linearity of the motion model. The observation function Jacobian H_{x_k} is the partial derivative of the observation function with respect to the state vector χ. This calculation is shown in its generalised form in Equation (5.85).

$$H_{x_k} = \frac{\partial h}{\partial \chi}(\widehat{\chi}_k^-) \tag{5.85}$$

The final representation of H_{x_k} is shown in Equation (5.86), where the matrix has been simplified for a clearer display. The component details of H_{x_k} are also shown below.

$$H_{x_k} = \boldsymbol{I_{6\times 6}} \tag{5.86}$$

The final metric required to be defined within the observation model is the observation noise covariance matrix represented by R. The matrix R is comprised of the variances of each component within the observation vector Z_k. The earlier presented section where the LiDAR odometry algorithm was assessed to study its error behaviour will be utilised within the construction of R as shown in Equation (5.87), where the subscript L denotes its origination from the LiDAR.

$$R = \begin{bmatrix} \sigma_{x_L}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{y_L}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{z_L}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\theta_L}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\phi_L}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\psi_L}^2 \end{bmatrix} \tag{5.87}$$

Where;

- σ_{x_L} - 1.1×10^{-3} (m)
- σ_{y_L} - 6.2×10^{-3} (m)

- σ_{z_L} - 8×10^{-3} (m)
- σ_{θ_L} - 2.75×10^{-2} (deg)
- σ_{ϕ_L} - 2.74×10^{-2} (deg)
- σ_{ψ_L} - 4.8×10^{-3} (deg)

5.7.3 Conclusions

In conclusion, Section 5.7 presents the formulation of the 6DOF-EKF as a solution for unmanned system state estimation in 3D. The 6DOF-EKF followed on from the R-EKF designed in Chapter 4 using an identical foundational kinematic model, however, incorporating an alternative observation source in the form of a laser scanning system. The 6DOF-EKF differs from the R-EKF in relation to the dimension of state estimation, where the R-EKF estimates motion in 2D and the 6DOF-EKF in 3D. The 6DOF-EKF executes a sensor fusion process utilising a kinematic model of the system in combination with quadrature encoders, accelerometers and gyroscopes to provide an initial predicted estimation, this is then updated using an ES-KF as a sub-system in combination with observations acquired from a LiDAR system. A final overview of the structure of the 6DOF-EKF is presented below in Algorithm 4.

5.8 Conclusions

In conclusion, Chapter 5 presented the background theory and operation of LiDAR systems and the corresponding state of the art within 6DOF state estimation for both indoor and outdoor environments concerning mobile robotic systems. It then expanded upon 6DOF motion estimation and demonstrates how it is possible to calculate this information from raw point cloud datasets acquired from a LiDAR sensor. This then led to the design and formulation of a 6DOF state estimation system through the utilisation of an EKF. Commercial and state of the art indoor positioning systems such as the Pozyx UWB network presented in Chapter 4 neglect the use of probabilistic state estimation and operate solely from sensor measurements alone. Therefore, the 6DOF-EKF designed in Chapter 5 is an alternative and improved solution to robotic system positioning. The main conclusions gathered from Chapter 5 are listed below.

Algorithm 4 Six Degree of Freedom EKF Structure

Prediction:

$u_k = [\gamma_{L_k}, \gamma_{R_k}, \omega_k]$

$\widehat{\chi}_k^- = f(\widehat{\chi}_{k-1}, u_k)$

$G_{x_k} = \frac{\partial f}{\partial x}(\widehat{\chi}_k^-, u_k)$

$G_{u_k} = \frac{\partial f}{\partial u}(\widehat{\chi}_k^-, u_k)$

$Q_k = G_{u_k}\ U\ G_{u_k}^T$

$\widehat{P}_k^- = G_{x_k}\widehat{P}_{k-1}G_{x_k}^T + G_{u_k}QG_{u_k}^T$

Correction:

[x, y, ψ] = 2DFeatureMatch($\boldsymbol{P_{k-1}}, \boldsymbol{P_k}$)

$[\theta, \phi, z] = ES\text{-}KF\ (\alpha_k, \omega_k)$

$Z_k = [x, y, z, \theta, \phi, \psi]^T$

$\widehat{Z}_k = h(\widehat{\chi}_k^-)$

$H_{z_k} = \frac{\partial h}{\partial x}(\widehat{\chi}_k^-)$

$K_k = \widehat{P}_k^- H_{z_k}(H_{z_k}\widehat{P}_k^- H_{z_k}^T + R)^{-1}$

$Y = Z_k - \widehat{Z}_k$

$\widehat{\chi}_k = \widehat{\chi}_k^- + K_k Y$

$\widehat{P}_k = (I - K_k H_{z_k})\widehat{P}_k^-$

if ! *measurement_is_available* **then**

    do **Prediction**

else

    do **Correction**

end if

- 6DOF motion of an unmanned system can be extracted from raw LiDAR point cloud output.
- Accelerometer and gyroscopic output are accompanied with slight biases which can affect overall roll and pitch angle estimations.
- Gyroscopic output estimations for all three angles drastically experience drift when native bias is unaccounted for.

- A fusion process within an ES-KF using raw accelerometer and gyroscopic output provides a more stable orientation estimation compared to using the sensors as a standalone.
- ES-KF orientation output and quadrature wheel encoder estimations have been integrated with a kinematic motion model of the unmanned system for state prediction ahead of time.
- Observations in 6DOF acquired from the LiDAR are used as an update source which is implemented post prediction. This forms the general structure of an EKF.

The following content presented in Chapter 6 is where the analysis is presented concerning the deployment of both positional estimation methods presented in Chapter 4 and Chapter 5. The R-EKF and the 6DOF-EKF are tested in a series of experimental procedures and are compared to ground truth metrics which enables the calculation of residual errors.

Part III

Algorithm Validation

Chapter 6

Quantitative Analysis

Chapter 6 presents a comparison study between the probabilistic state estimation algorithms R- EKF and 6DOF-EKF designed in Chapter 4 and Chapter 5 respectively. The comparison study includes the execution of two experimental procedures. The first is conducted within an open and wide experimental environment on flat terrain, where the unmanned system is controlled to execute a set of experimental trajectories, and the results gathered from each algorithm are compared to the ground truth metrics and current state of the art systems. The second experimental procedure is conducted in a more isolated area. This procedure includes the unmanned system exploring terrain which experiences a change in inclination, this is conducted to validate the positioning techniques able to estimate the change in altitude, roll angle and pitch angle motion components (z, θ, ϕ), where the aforementioned components of motion are subjected to no change when operating on flat terrain. The ground truth metrics for translational and rotational components of motion are acquired using the RTS and the current commercially deployed Bosch IMU respectively. The inclusion of ground truth metrics enables algorithm performance quantification for quality of comparison and justification of which system is superior. The structure of Chapter 6 is as follows;

- Introduction to Ground Truth Metrics - Provides an insight into the sources of information used as ground truth metrics to enable a quantitative analysis for comparison between all studied techniques. This section is split into two sub-sections which cover the ground truth instruments for both translational and rotational components of motion.
- Methodology Flat Terrain - Provides an overview of all components that will be

compared and analysed to assess which provides the superior performance. Also presents the experimental validation methodology for the flat terrain environment that was executed to compare the output of all positioning systems. This includes an overview into the example trajectories conducted by the unmanned platform.

- Methodology Inclined Terrain - Provides an overview of all components that will be compared and analysed to assess which provides the superior performance in such an environment. Also presents the experimental validation methodology for the inclined terrain environment. This methodological approach was undertaken to test the unmanned system in isolated areas. Components of motion Cartesian Z (altitude), roll and pitch angles (z, θ, ϕ) only are considered.

- Results Flat Terrain - Assesses and analyses the outputs from all components providing position estimations of the unmanned system against corresponding ground truths for the flat terrain procedure.

- Results Inclined Terrain - Assesses and analyses the outputs from the components of motion related to the inclined terrain experimental methodology; corresponding ground truths for the flat terrain procedure.

- Scoring Table - Summarises the results within a scoring table demonstrating the characteristics of each experimented solution as well as the ground truth metrics.

- Conclusions - Concludes Chapter 6 with a discussion on the performance of the positioning systems with respect to ground truth metrics. Also executes a weighting process against each system for various types of categories of performance, concluding with the superior system.

6.1 Ground Truth Metrics

Section 6.1 presents an overview of the reference frames of the information sources used as the ground truth metrics. The importance of a ground truth enabled a clear quantitative analysis concerning algorithm performance. Also, it further enabled a comparison study between each technique. The instruments used as the ground truth for both translational and rotational components of motion are outlined in Section 6.1.1 and Section 6.1.2

respectively.

6.1.1 Translational Components

The estimations concerning the translational components produced from the various types of positioning techniques were compared against the RTS to quantify performance. As presented earlier in Chapter 4, the RTS can be deployed with an active tracking feature. Once the RTS is locked onto a prism on-board the unmanned system, it can provide location estimations with millimetre accuracy. The operating co-ordinate frame of the RTS is shown in Figure 6.1.

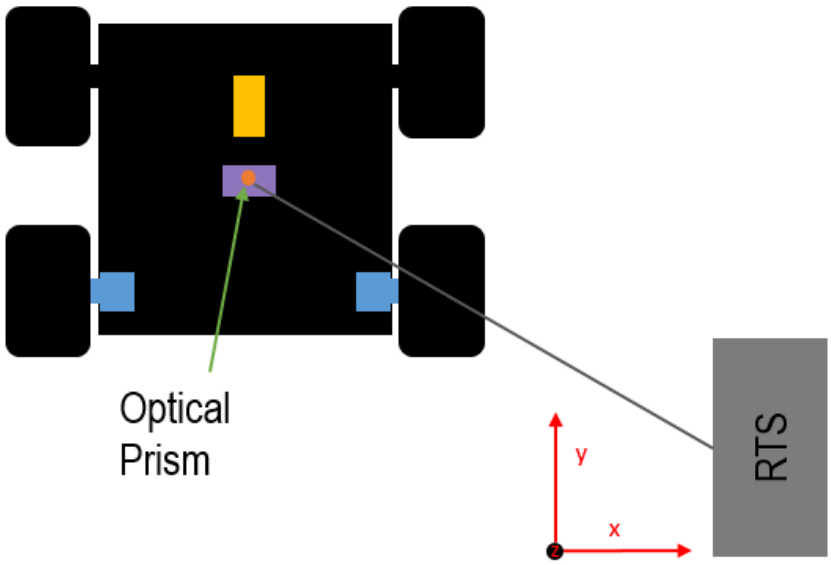


Figure 6.1: Translational Ground Truth Metric Reference Frames

6.1.2 Rotational Components

The instrument used as the ground truth metric for analysing the rotational components of motion was an IMU designed by Bosch. The Bosch IMU deploys a 9DOF adaptive filtering technique to estimate the orientation of the body in either quaternion or Euler angle format. In order for the output from the Bosch IMU to be compared to the rest of the sensors on-board the unmanned system, it was paramount that it was mounted with its orientation operating within the same reference frame. Figure 6.2 shows the mounting structure of the Bosch IMU where it maintains the forward facing direction within the Y axis identical to the other on-board sensory systems.

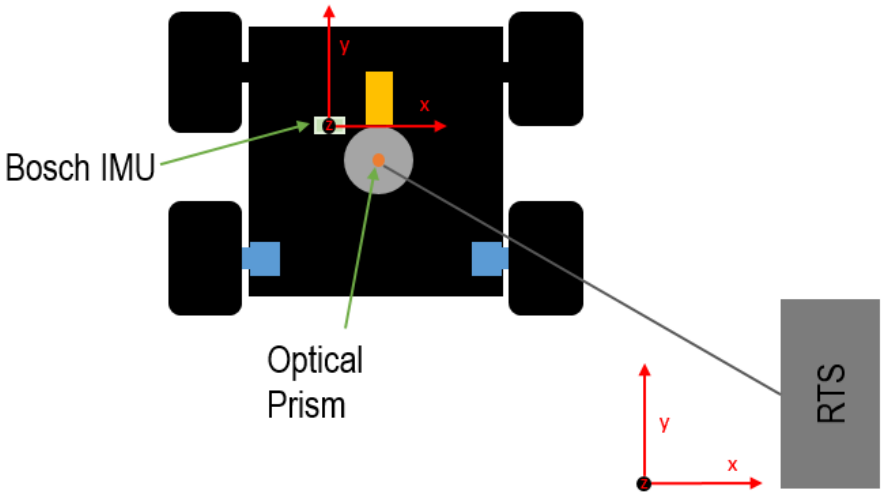


Figure 6.2: Translational Ground Truth Metric Reference Frames

6.1.3 Conclusions

Section 6.1 introduced the instruments and sources of sensory information used as ground truth metrics for the experimental procedures. The ability to introduce a flexible ground truth enables quantitative analysis between the different deployed techniques, and presents the opportunity to carry out a comparison study. The next section introduces the system level architecture and how all outputs acquired from systems on-board the unmanned vehicle integrate through ROS.

6.2 System Level Architecture

Section 6.2 presents the system architecture used for the experimental procedure. It covers the integration between the sensors operating on-board the unmanned system with them operating off-board and how time synchronisation issues were resolved.

6.2.1 Data Transparency

The transparency between the data sources utilised for the validation and comparison experiment was paramount to ensure that a structured and unbiased study was conducted. Each sensor that operated on -board the unmanned system possessed its own physical connection to the on-board NVDIA Jetson TX1 with the Orbitty shield. Acquiring the data outputted from each sensor required the design of custom software that interacted

with the sensor and organised the data that it was providing. However, in order to conduct a full comparison study, the data from all the individual sensors had to be time-synchronised. The method chosen to achieve this time synchronisation was through ROS. ROS is a meta-operating system which functions through transparent message publishment and subscription through a group of inter-communicating nodes within a network. Each node communicates with what is known as the ROS Core, which acts as the central hub for full data transparency between nodes on the established ROS network. For the case of the sensing systems on board the unmanned platform, time synchronisation was a fairly simplistic process as these sensors were connected to the TX1, which acted as the hub of the whole unmanned platform. The TX1 possesses its own relative clock, therefore the messages published from the nodes linked to each individual sensor were tagged with a time-stamp with reference to the TX1s' relative clock. Therefore, all nodes operating on the TX1 had a common time source. However, the TX1 does not possess an RTC, therefore if the TX1 is disconnected from an online network, its clock will reset to a factory default date as it has no method to connect to a time server to update the clock. This proved to be an issue as the RTS is the only instrument within the experimental process that is not operating on board the unmanned system, it therefore has no means to be connected physically to the TX1 to share the same clock reference as the other sensing instruments. Also, as the RTS is a commercial instrument, a custom ROS serial driver was designed to read and parse the output acquired from the RTS and was set to operate on a laptop ground station. The initial issue with this process is that the laptop possesses an RTC, so if disconnected from a network, it can still maintain a correct time source, which is contrasting to that of the TX1. The overall and underlying issue is that the ground station and the TX1 had misaligned clocks and the outputs from the RTS would be unsynchronised with the rest of the experimental instruments. Therefore, the RTS readings acquired from the ground station were then programmed to be published onto the ROS network, and with the ground station and the TX1 sharing the same network via a WAP, the RTS readings from the ground station can be acquired by the TX1 using the ROS message data transparency functionalities. This therefore solves the time synchronisation issues. The whole system architecture with each sensor, corresponding nodes and the ground station set-up can be seen in Figure 6.3.

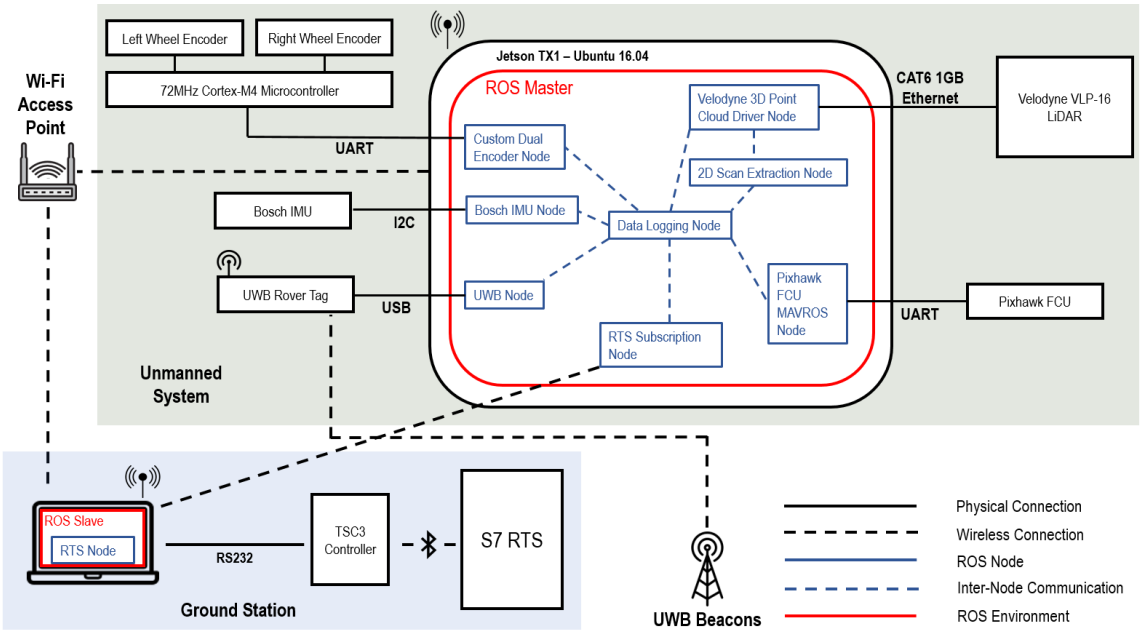


Figure 6.3: Lower Level System Architecture

6.2.2 Conclusions

In conclusion, Section 6.2 introduced the system level architecture implemented on-board the unmanned system during experimental procedures. It demonstrated how the deployment of a master-slave pair both functioning using the ROS meta-operating system enables time synchronisation between all sources of information. The following content within Section 6.3 will begin the presentation of the experimental methodologies used to compare each state estimation technique.

6.3 Methodology Flat Terrain

Section 6.3 presents the methodology concerning the validation experimental procedure of the flat terrain environment. With the UWB system requiring that static beacons be situated within the experimental environment, this process is explained within its own section as it provides localisation services. Other systems such as the 6DOF-EKF and raw odometry provide estimations within the relative body frame of the unmanned system, therefore they do not require reference points within the environment. The lists below present the techniques that were deployed within the validation experimental study. The two lists are split into two categories of comparison, the translational aspects of motion and the rotational aspects of motion. The first component of each list presents the information

source used as the ground truth metric for that specific category. This enables the coverage of estimation concerning full 6DOF. It is to be noted that raw odometry is estimation from the proprioceptive quadrature wheel encoders alone, the output of this technique that is included within the validation study is the initial prediction stage of the R-EKF and 6DOF-EKF alone, without the update from external sensor observations.

Translational

- RTS (Ground Truth)
- Native UWB Positioning Algorithm
- R-EKF
- Odometry
- 6DOF-EKF

Rotational (Heading Only)

- Bosch IMU (Ground Truth)
- 6DOF-EKF
- R-EKF
- Odometry

6.3.1 Experimental Environment

The environment chosen to host this experimental procedure had to be one with a sufficient amount of space to fully test the flexibility of each positioning technique. Therefore, an indoor open testing space was utilised as shown in Figure 6.4. This experimental environment enabled the RTS to have robustness in its tracking capabilities and to provide a ground truth metric throughout the procedure. From observing Figure 6.4, it can be seen that the terrain is of a flat nature and does not experience any deviations in elevation. Therefore, when validating the Z Cartesian and the roll and pitch components of 6DOF motion, there is an expectation that these components will not change. The nature of the experimental environment terrain can be seen in Figure 6.5, where the dataset represents the relative altitude changes of the terrain acquired from the RTS as the unmanned system explored the environment.

Figure 6.4: Experimental Environment

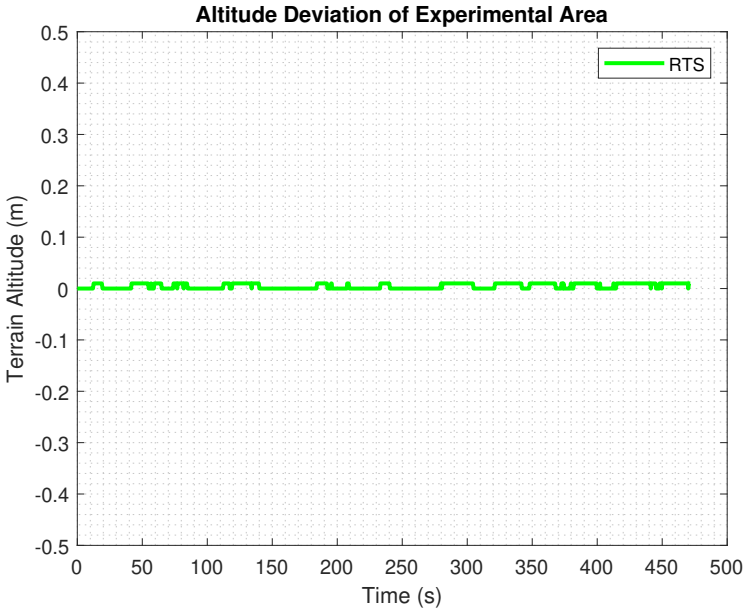


Figure 6.5: Altitude Deviation of the Flat Terrain Experimental Environment

6.3.1.1 RTS Location

The calibration point of the RTS is a vital stage required to be conducted prior to commencing the experimental procedure. It acts as the reference point for all positioning techniques that were validated. The procedure conducted to calibrate the RTS presented in Chapter 4 is consistently implemented here within the validation experiment in Chapter 6, where a back-sight measurement is taken to calibrate the co-ordinate system that the RTS will use. Additionally, the height of the rotating head of the RTS is measured to provide

a reference for detection of relative altitude change of the unmanned platform. The initial 3D reference point set for the RTS is shown in Equation (6.1), therefore all measurements acquired from the RTS are in relation to $RTS_{\boldsymbol{P}}$.

$$RTS_{\boldsymbol{P}} = [0\ 0\ 0]^T \tag{6.1}$$

It is to be noted that $RTS_{\boldsymbol{P}}$ can be a customised value as it is providing a relative ground truth metric within this thesis. If the RTS was used to provide ground truth services for sensors that are required to operate in a global frame of reference such as GNSS, $RTS_{\boldsymbol{P}}$ would be set as a point within WGS84 or the Galileo Terrestrial Reference Frame (GTRF) depending upon which terrestrial reference system the GNSS unit utilises. This usually involves extra components to be integrated within the RTS such as Real-Time Kinematic (RTK) GNSS receivers to provide precise geo-localisation features, however, such features are not capable to operate effectively within indoor environments.

6.3.1.2 UWB Network Structure

As presented in Chapter 4 for the range error characterisation of the UWB sensor network, the RTS was utilised to acquire the true locations of the UWB anchors. This procedure was carried into the validation experimental process presented here in Chapter 6. The UWB beacons were scattered in an disorganised fashion to provide full coverage of the experimental environment. The RTS was then used to acquire the true position of each beacon in relation to the calibrated RTS position presented in Section 6.3.1.1. This enabled the observations acquired from the UWB sensor network to be within the same operating reference frame as the RTS. The set-up of the static UWB beacons is shown in Figure 6.6.

6.3.1.3 Experimental Trajectories

The trajectories conducted as a part of the experimental procedure were designed with various levels of difficulty. Four main experimental trajectories were chosen to thoroughly test each aspect of the algorithm's estimation procedures. Figure 6.7 demonstrates the nature of the four experimental trajectories using the acquired RTS data. As seen in Figure 6.7, the trajectories increase in difficulty in order to test how each algorithm handles aspects such as rapid changes in heading.

Each of the four trajectories was conducted with three repetitions, a total count of 12

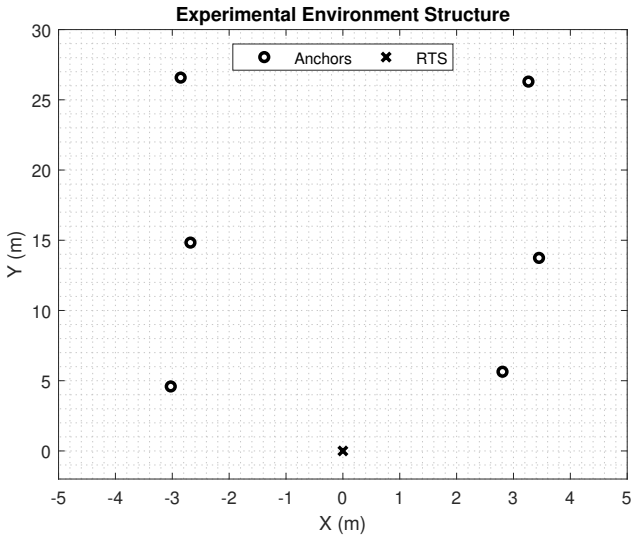


Figure 6.6: UWB Anchor & RTS Set-up

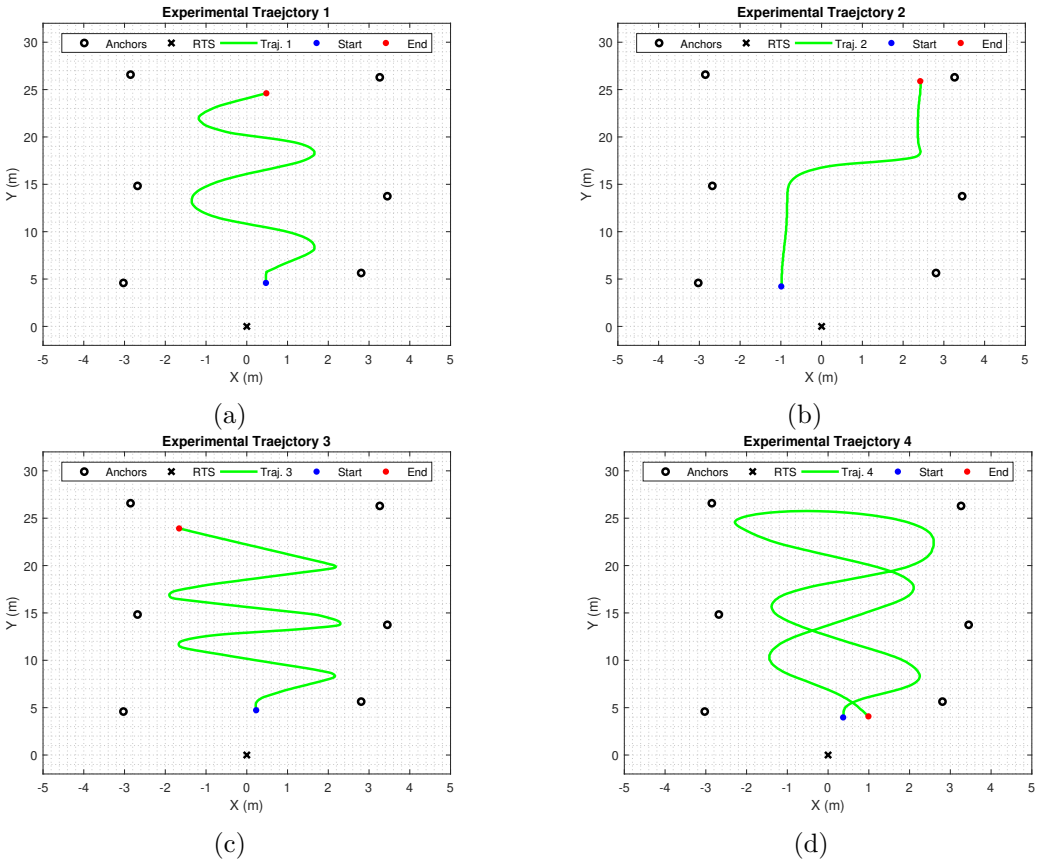


Figure 6.7: Experimental Trajectories (a) Experiment 1 (b) Experiment 2 (c) Experiment 3 (d) Experiment 4

experimental tests. For each trajectory, the estimated components of motion were compared against the corresponding ground truth metrics through a quantitative analysis.

6.4 Methodology Inclined Terrain

Section 6.4 introduces the methodology conducted concerning the operation on inconsistent terrain. The underlying reason for the inclusion of this experiment is that during the flat terrain experiment introduced in Section 6.3, the motion components Z, roll and pitch (z, θ, ϕ) experience no change due to the nature of the terrain. Therefore, an experiment conducted within an area that possessed terrain inconsistency was executed. This experiment only considered the techniques capable of estimating z, θ and ϕ. These techniques tested during this experiment are listed below for translational and rotational components.

Translational (Z Only)

- RTS (Ground Truth)
- 6DOF-EKF

Rotational (Pitch Only)

- Bosch IMU (Ground Truth)
- 6DOF-EKF

6.4.1 Experimental Environment

Section 6.4.1.1 presents an overview of the experimental environment utilised to test the algorithm in environments where terrain elevation was inconsistent. The reason for executing such experimental procedure was to assess how the algorithm detects the unmanned system's increase in elevation. Figure 6.10 provides a graphical overview of the environment, the unmanned system within the environment and the area of terrain that is seen to incline in relation to the flat terrain where the unmanned system originated from. The key aspects to the inclined terrain experimental procedure were to compare the 6DOF-EKF output for Cartesian Z position against the RTS and the rotational pitch angle component against the ground truth of the commercial Bosch IMU.

The unmanned system was configured to execute a trajectory which initially started on a flat section of the terrain and then progressed onto the incline. Due to the configuration of the unmanned system's frame of reference, as it embarks upon the incline, it would

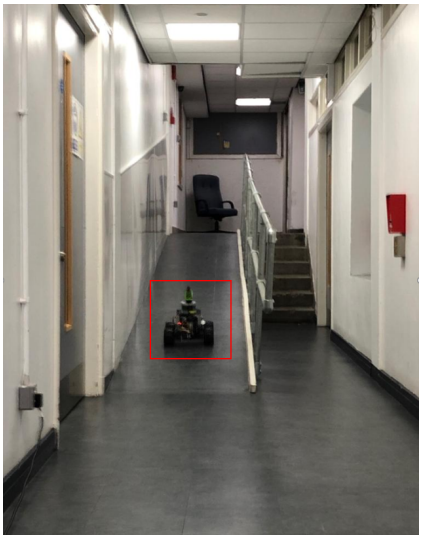


Figure 6.8: Terrain inclination during experimental procedure

experience a change in its pitch angle as shown in Figure 6.9. The on-board commercial Bosch IMU would provide constant updates concerning this angle change and the 6DOF-EKF would also estimate the angle, a quantitative analysis was executed to compare the output. Additionally, with the RTS deploying its active tracking capabilities, it was possible to extract the unmanned system's change in altitude in relation to its starting position, this enabled a quantitative analysis concerning the RTS and 6DOF-EKF outputs for the Cartesian Z position.

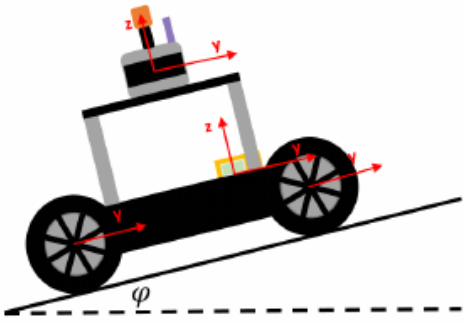


Figure 6.9: Pitch angle change

The nature of the experiment consisted of 5 repetitions of the unmanned system executing the trajectory along the inclined terrain. Each trajectory output for Z position and pitch angle were compared to the corresponding ground truth metric for RMSE and mean bias error (MBE).

6.4.1.1 RTS Location

The location of the RTS was set to provide a clear line of sight to the on-board miniature prism. As the robot and the RTS started from the same flat terrain, the relative altitude was equal and therefore the change when encountering the inclined terrain could be compared. Figure 6.10 shows the location of the RTS in relation to the inclined patch of terrain and the unmanned system as it travels up the incline. It is clear that there is always a clear line of sight to the robot, a factor which the active tracking capabilities of the RTS requires.

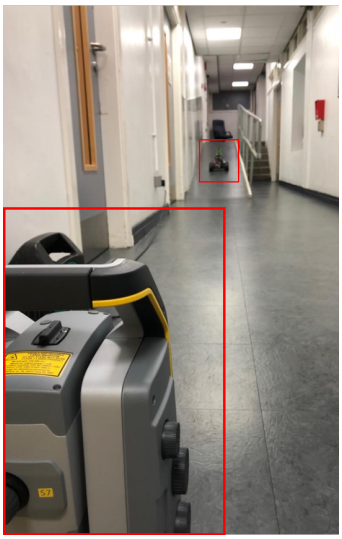


Figure 6.10: RTS location during experimental procedure

6.5 Results Flat Terrain

Section 6.5 presents the results acquired from the experimental procedure conducted in the flat terrain environment. The results are separated into four sections which cover the results acquired for the 2D pose (x, y, ψ) from each type of trajectory. These can be seen in Sections 6.5.1 to 6.5.4, where each of the sections provides both tabulated and graphical results of the 2D poses compared to the ground truth metrics.

6.5.1 Trajectory 1

Section 6.5.1 presents the results acquired for each experimental repetition of Trajectory 1. It compares the motion components of Cartesian position x and y and the system's relative heading. The methods compared are raw odometry, the native algorithm deployed by the UWB system, the range based EKF and the 6DOF-EKF.

Figure 6.11 presents the results acquired graphically from repetition 1 of Trajectory 1. From first visual assessment of the 2D trajectory comparison shown in Figure 6.11a, it can be seen that the output from raw odometry only produces a smooth trajectory estimate with noise present. This demonstrates that the gyroscope used to calculate the odometry trajectory has a well accounted bias. However, it is clearly evident that the trajectory is seen to experience high levels of drift. This is both a combination between the error in encoder estimations as well as raw gyroscopic heading calculations. This drift is also identifiable when isolating the components of motion for X and Y shown in Figure 6.11b and Figure 6.11c respectively, where the motion components for raw odometry (red) are seen to initially follow the ground truth (green) during the early stages of the trajectory but drift catastrophically over time.

The native algorithm deployed by the commercial UWB system (blue) is seen to not experience drift over time as opposed to the raw odometry estimations, this is due to the consistent reference points in the static anchor transceivers within the environment to prevent drifting. However, the native UWB experiences erroneous measurements that result in noisy position estimations, this can be clearly identified in Figure 6.11a and for the native UWB algorithm output in X position within Figure 6.11b. The R-EKF which uses the raw individual range measurements from each anchor provides a trajectory estimation with a reduction in noise compared to the native UWB. The R-EKF demonstrates effectiveness in combining the smooth yet drift prone odometry with noisy yet positionally accurate raw UWB range measurements to provide an output which improves upon the native UWB algorithm. The consistent outlier detection operation within the R-EKF identifies and mitigates the effects of erroneous measurements. This can be identified in Figure 6.11a and Figure 6.11b where the R-EKF (magenta) ignores and reduces the effects of noisy UWB measurements.

The results acquired from the 6DOF-EKF, which included the fusion of odometry and LiDAR produced the superior results for Trajectory 1. It is evident that the 6DOF-EKF for components X and Y (black) are both highly precise and accurate in comparison to the ground truth of the RTS as shown in Figure 6.11b and Figure 6.12b. The 6DOF-EKF demonstrates stable measurements in comparison to the other UWB based methods as it is less prone to communication errors due to not requiring external wireless communication with environmental beacons.

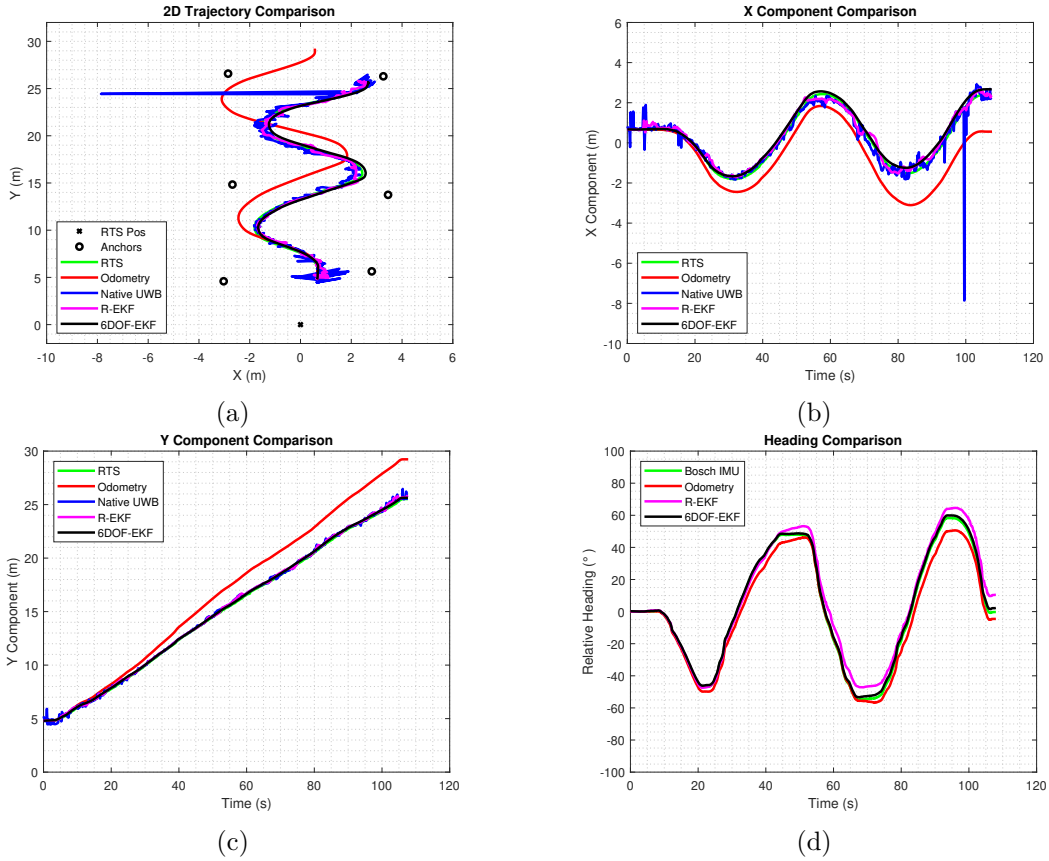


Figure 6.11: Components of motion comparison - Trajectory 1 (a) 2D Trajectory (b) Cartesian X (c) Cartesian Y (d) Heading

The results for Trajectory 1 then have been tabulated to provide numerical reference. These results can be seen in Table 6.1 for Cartesian aspects of motion and Table 6.2 for the heading comparison. When focusing on Table 6.1, it is evident that for all three repetitions of Trajectory 1, raw odometry is seen to produce the highest metrics for RMSE and MBE in comparison to the other three techniques. It is also interesting to focus on the output of the R-EKF in comparison to the traditional UWB technique. The R-EKF produces significantly lower RMSE than the native UWB for both X and Y position estimations for all three repetitions, demonstrating an improvement upon a current commercially acquirable system in this environment. It is also seen to reduce the MBE for both X and Y for all repetitions apart from the first repetition, where MBE for the X motion component from the native UWB produces a lower MBE in comparison to the R-EKF.

Table 6.2 presents tabulated results concerning the quantitative comparison of the heading estimations for each technique. This can be graphically observed in Figure 6.11d. It is clear from Figure 6.11d that the heading estimations from the 6DOF-EKF perform

Table 6.1: Cartesian Component Analysis Trajectory 1

Metric (m)	**Odometry**	**Native UWB**	**R-EKF**	**6DOF-EKF**
RMSE (x)	2.9674	0.2934	0.2810	0.0643
	0.5897	0.4211	0.1916	0.1078
	2.2374	0.4001	0.2809	0.1002
RMSE (y)	0.6294	0.1475	0.1176	0.0629
	1.0885	0.1660	0.1181	0.0402
	0.8796	0.2130	0.1609	0.0472
MBE (x)	3.1926	0.1970	0.2270	0.0191
	0.8760	0.2065	0.1761	0.1682
	1.9328	0.2401	0.2296	0.1302
MBE (y)	0.7624	0.1569	0.1439	0.1098
	1.7341	0.1505	0.1355	0.1016
	1.2791	0.1971	0.1803	0.0335

much better in comparison to the odometry and R-EKF, where the odometry alone is seen to drift in certain areas without the inclusion of supporting sensors. The tabulated results agree with those shown in Figure 6.11d, where the 6DOF-EKF produces the lowest RMSE and MBE at sub 2° readings. The odometry alone proved to provide the worst heading estimations, this is expected as using the yaw rate gyroscope as a standalone may include adapting gyroscopic biases and noise levels.

Table 6.2: Heading Analysis Trajectory 1

Metric (∘)	**Odometry**	**R-EKF**	**6DOF-EKF**
RMSE	15.9626	13.4693	1.0211
	3.0657	3.8806	0.6523
	14.7829	9.3312	0.8990
MBE	22.9052	21.0030	1.3192
	3.8234	3.7821	1.2312
	20.7210	16.2325	1.4198

6.5.2 Trajectory 2

Section 6.5.2 presents the results acquired for each experimental repetition of Trajectory 2. Trajectory 2 proved to be slightly more complex than Trajectory 1 as there are segments where the unmanned system manoeuvred along a straight path. This tested how each technique drifted when experiencing no heading change. Figure 6.12 presents the results acquired from Trajectory 2. It is evident from initially observing Figure 6.12a, which graphically compares the 2D trajectory, that the odometry estimation is seen to experience high levels of drift. Although the trajectory is of a similar nature to the ground truth metric of the RTS, it demonstrates that it overshoots the segments of the trajectory where a turning manoeuvre occurs. This may be due to wheel slippage produced by the quadrature wheel encoders, further resulting in the over-estimation of the change in forward facing linear displacement. This is also seen in Figure 6.12b, Figure 6.12c and Figure 6.12d for all three components of motion for odometry, where the red plot is seen to deviate from the ground truth and the outputs acquired from the other techniques. The odometry estimations concerning the heading motion component show that when the unmanned system travelled along the straight segment of the trajectory, the odometry heading was seen to drift over time. This effect is an example of why gyroscopic outputs usually need to be combined with other sensors to stabilise this drifting characteristic.

From observing and comparing the output of the native UWB algorithm and the R-EKF, it is observable from assessing the 2D trajectory output in Figure 6.12a that the R-EKF significantly reduces the standard deviation of position estimation. The native UWB is seen to provide slightly volatile measurements in areas further away from the location of the RTS. However, the R-EKF is seen to maintain a smooth trajectory estimation and this demonstrates the effectiveness of the live outlier detection and mitigation process within the R-EKF. This is also clearly observable in Figure 6.12b for the X motion component, where the native UWB algorithm experiences outlier measurements and the R-EKF is seen to adjust its estimation accordingly. Similar to Trajectory 1, the 6DOF-EKF outputs provide the superior state estimations when compared to the ground truth metric. It shows how the inclusion of motion estimation using a LiDAR system is both accurate and precise. However, in the case of Trajectory 2, the 6DOF-EKF estimation demonstrates small levels of drift inaccuracies as it approaches the final stages of the trajectory. This can be mainly

observed in Figure 6.12b, where the 6DOF-EKF data plot is seen to deviate from the RTS during the final stages of the experiment. The final error metrics for the Cartesian components of motion for Trajectory 2 are displayed in Table 6.3.

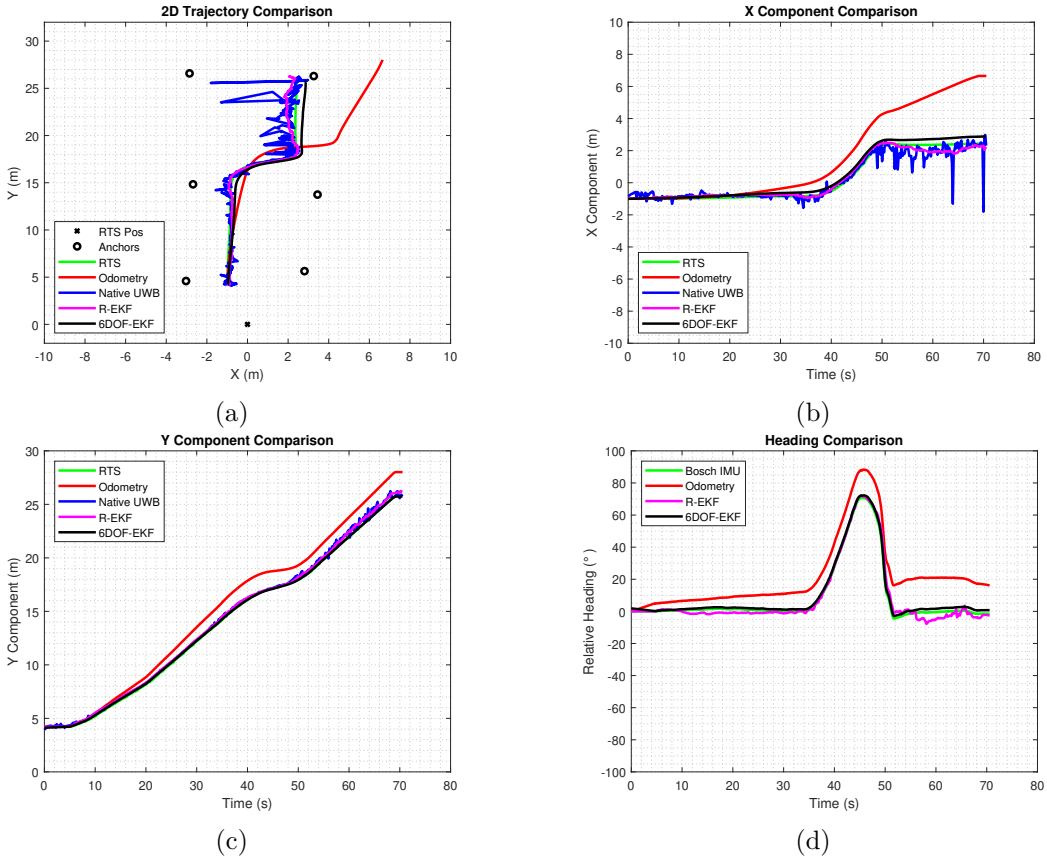


Figure 6.12: Components of motion comparison - Trajectory 2 (a) 2D Trajectory (b) Cartesian X (c) Cartesian Y (d) Heading

When analysing the heading outputs acquired from each technique for Trajectory 2, it can be seen from observing Figure 6.12d that the odometry alone experiences drift over time when the unmanned system navigates along a straight segment of the trajectory. This, however, is corrected from the output produced by the R-EKF. Although the R-EKF is not as stable in comparison the 6DOF-EKF and the ground truth metric, it is shown to significantly reduce the drifting aspect when the unnamed system is embarking upon the straight parts of the trajectory. Additionally, the R-EKF also estimates the change in heading very well and follows a near identical pattern to that of the Bosch IMU (ground truth) and 6DOF-EKF. The final results concerning the heading component for all three repetitions of Trajectory 2 are tabulated in Table 6.4, where it shows that the 6DOF-EKF performed the best in relation to overall RMSE and MBE for all of the techniques.

Table 6.3: Cartesian Component Analysis Trajectory 2

Metric (m)	**Odometry**	**Native UWB**	**R-EKF**	**6DOF-EKF**
RMSE (x)	1.3543	0.4211	0.2111	0.1288
	2.2050	0.3819	0.1849	0.2300
	1.1449	0.6579	0.2581	0.3751
RMSE (y)	0.6153	0.1827	0.1218	0.0621
	0.3568	0.1913	0.1510	0.0719
	0.6995	0.1979	0.1349	0.1218
MBE (x)	1.3851	0.2362	0.1696	0.2392
	2.1844	0.2432	0.1808	0.3385
	1.2769	0.2949	0.2192	0.6254
MBE (y)	1.2767	0.2158	0.2031	0.0237
	0.7752	0.2153	0.1803	0.1079
	1.4493	0.2199	0.2028	0.0577

Table 6.4: Heading Analysis Trajectory 2

Metric (∘)	**Odometry**	**R-EKF**	**6DOF-EKF**
RMSE	6.0385	2.4082	0.5972
	11.7792	8.6419	0.6309
	4.5899	6.6289	0.9485
MBE	13.4283	0.6675	1.3510
	20.2421	14.0811	1.0060
	3.5835	3.0056	1.6859

6.5.3 Trajectory 3

Section 6.5.3 presents the results acquired concerning an iteration of experimental Trajectory 3. Trajectory 3 was implemented to assess how each technique dealt with consistent changes of the unmanned system's heading. The graphically presented results for Trajectory 3 are shown in Figure 6.13. Firstly, when observing the resulting 2D trajectory plots for each technique shown in Figure 6.13a, at points during the trajectory execution the unmanned system executes a turning manoeuvre. During these turning procedures, the native UWB algorithm is seen to experience noise within position estimations. This

demonstrates that as the unmanned system reduces its speed to execute the turning procedure, the native UWB algorithm's output increases in levels of positional uncertainty in comparison to segments of the trajectory where the unmanned system is not turning. This can also be observed in Figure 6.13b for the X component around the minimum and maximum points. This, however, is seen to be reduced when raw UWB range measurements are used as a source of information with the R-EKF. It can be seen that the output from the R-EKF is visually a lot smoother and evidently reducing the uncertainty in estimation compared to the native UWB. In comparison to both the native UWB algorithm and the R-EKF, the results acquired concerning the 6DOF-EKF were seen to visually produce the most accurate and precise results as it previously did for Trajectories 1 and 2. However, during the final turning manoeuvre of Trajectory 3, the 6DOF-EKF experiences slight inaccuracies in heading estimations, which consequently affects its estimation in X. This can be seen in both Figure 6.13b and Figure 6.13d when comparing the 6DOF-EKF against the ground truth metrics (green). Additionally, it can be seen from all sub-figures within Figure 6.13 that like the results acquired in Trajectory 1 and 2, odometry as a form of relative position estimation experiences catastrophic position and orientation estimation errors when used as a standalone.

The final results for all iterations of Trajectory 3 for each technique concerning the motion components of the 2D pose can be seen in Table 6.5. The numerical results concerning the three experimental iterations for Trajectory 3 show that for the RMSE of the X component, the R-EKF outperformed the three other techniques on two repetitions. This result, however, can not necessarily be determined by visually inspecting Figure 6.13. Additionally, this outcome was not seen in the previous two experimental trajectories, where the 6DOF-EKF was seen to produce the superior results. This shows that during circumstances where there is a significant level of heading change occurring, the sensor fusion output from the R-EKF can provide more stable motion estimations on the X axis. The R-EKF also produced lower MBE on the Y axis compared to the other techniques. Out of the 3 currently covered experimental trajectories, Trajectory 3 is the only trajectory where the 6DOF-EKF has not produced the best state estimation results out of the deployed techniques. Possibly showing that as the trajectories increase in complexity, the performance of the 6DOF-EKF is seen to slightly deteriorate.

Table 6.6 presents the results concerning the heading component of motion, where the

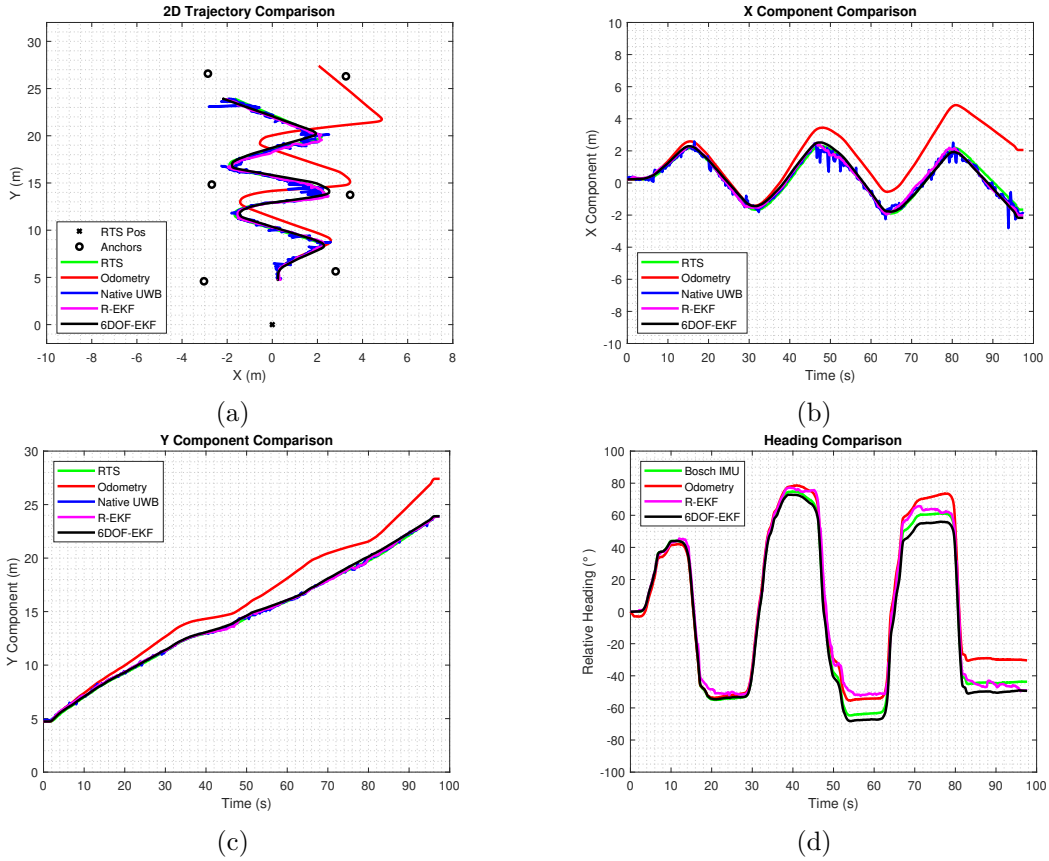


Figure 6.13: Components of motion comparison - Trajectory 3 (a) 2D Trajectory (b) Cartesian X (c) Cartesian Y (d) Heading

Bosch IMU was used as a means of a ground truth metric. These numerical results are to be discussed in conjunction with the plots shown in Figure 6.13d. It can be seen from observing Figure 6.13d that the heading estimations produced from the odometry technique alone is seen to be erroneous, especially in areas where the heading is constant during straight segments of the trajectory. This erroneous occurrence is overcome by the R-EKF and can be best observed during the final peak of the plots shown in Figure 6.13d. The 6DOF-EKF produced the most stable heading estimations in comparison to the ground truth metric. It can be seen to be both accurate and precise in most areas, however, there are areas which demonstrate possible drifting characteristics. This can be seen in the final stages of the trajectory, where the 6DOF-EKF overestimates the change in heading prior to the unmanned system embarking upon the final straight segment of the trajectory. Finally, from assessing both Figure 6.13d and the tabulated results from all three trajectory experimental iterations shown in Table 6.6, it can be seen that the R-EKF outperforms odometry as a standalone, however, the 6DOF-EKF was the optimal performing technique, showing that

Table 6.5: Cartesian Component Analysis Trajectory 3

Metric (m)	**Odometry**	**Native UWB**	**R-EKF**	**6DOF-EKF**
RMSE (x)	1.0866	0.2732	0.2330	0.2316
	0.4083	0.2376	0.2286	0.2507
	0.7716	0.2412	0.1781	0.3908
RMSE (y)	0.8642	0.1123	0.1017	0.0992
	0.8633	0.1166	0.0795	0.0619
	1.0192	0.1206	0.0939	0.1018
MBE (x)	1.3085	0.1968	0.1922	0.0353
	0.6225	0.1838	0.2014	0.0898
	1.2765	0.1735	0.1500	0.3846
MBE (y)	1.5951	0.1051	0.1041	0.1545
	1.4051	0.1356	0.1250	0.1348
	1.7227	0.1248	0.1191	0.1120

the use of a LiDAR in a feature-rich indoor environment improves heading estimation accuracy and precision.

Table 6.6: Heading Analysis Trajectory 3

Metric (∘)	**Odometry**	**R-EKF**	**6DOF-EKF**
RMSE	5.7624	4.8428	2.3882
	2.3470	1.3795	1.7356
	3.6872	5.1184	1.1509
MBE	6.3493	3.7297	2.7414
	3.5280	2.0005	2.3445
	7.5840	9.9117	1.517

6.5.4 Trajectory 4

Section 6.5.4 presents the results acquired from Trajectory 4, which was the final conducted experimental trajectory. Trajectory 4 was designed to simulate a traditional route taken through an indoor environment which was cluttered with obstacles. It was conducted to fully test each technique's ability to execute a trajectory of a complex nature and with the presence of ground truth metrics, also quantify the uncertainty of each technique. The graphical results acquired from Trajectory 4 are shown in Figure 6.14. Firstly, from observing Figure 6.14a, it can be seen that the use of the traditional odometry technique without the input of external sensing sources experiences a cumulative heading drift as well as an over estimation in translational motion. This demonstrates that the use of a gyroscope can experience drifting aspects when deployed as a standalone and not fused with other sources of information. Additionally, it can be seen from both Figure 6.14b and Figure 6.14c that the combination of gyroscopic drift and wheel slippage from the encoders hinders the accuracy of the Cartesian position estimations for the odometry technique as a standalone solution. As seen in the previous three experimental trajectories, the R-EKF provides a more improved position estimation in comparison to the native commercial UWB system's localisation algorithm. This is also demonstrated in Trajectory 4 where the R-EKF is seen to provide estimations with lower RMSE and MBE for both Cartesian components. This statement can be confirmed through visually observing both Figure 6.14b and Figure 6.14c and within Table 6.7. The performance of the 6DOF-EKF in Trajectory 4 proved to be the most effective technique in comparison to the traditional odometry, native UWB and the R-EKF in terms of precision and accuracy. This can be observed for all components within Figure 6.14. However, there are areas where the 6DOF-EKF produces slight erroneous estimations, specifically when the unmanned system navigates to its lowest position on the X axis in Figure 6.14b.

The final results acquired from trajectory 4 for the Cartesian component estimations from each technique are tabulated below in Table 6.7. This further confirms that for experimental Trajectory 4, the 6DOF-EKF was the most effective technique. Additionally, it also confirms that the R-EKF, which uses raw UWB range measurements, outperforms the native algorithm deployed by the commercial UWB system. Finally, Table 6.7 also confirms that deploying an odometry technique as a standalone solution lacks in precision

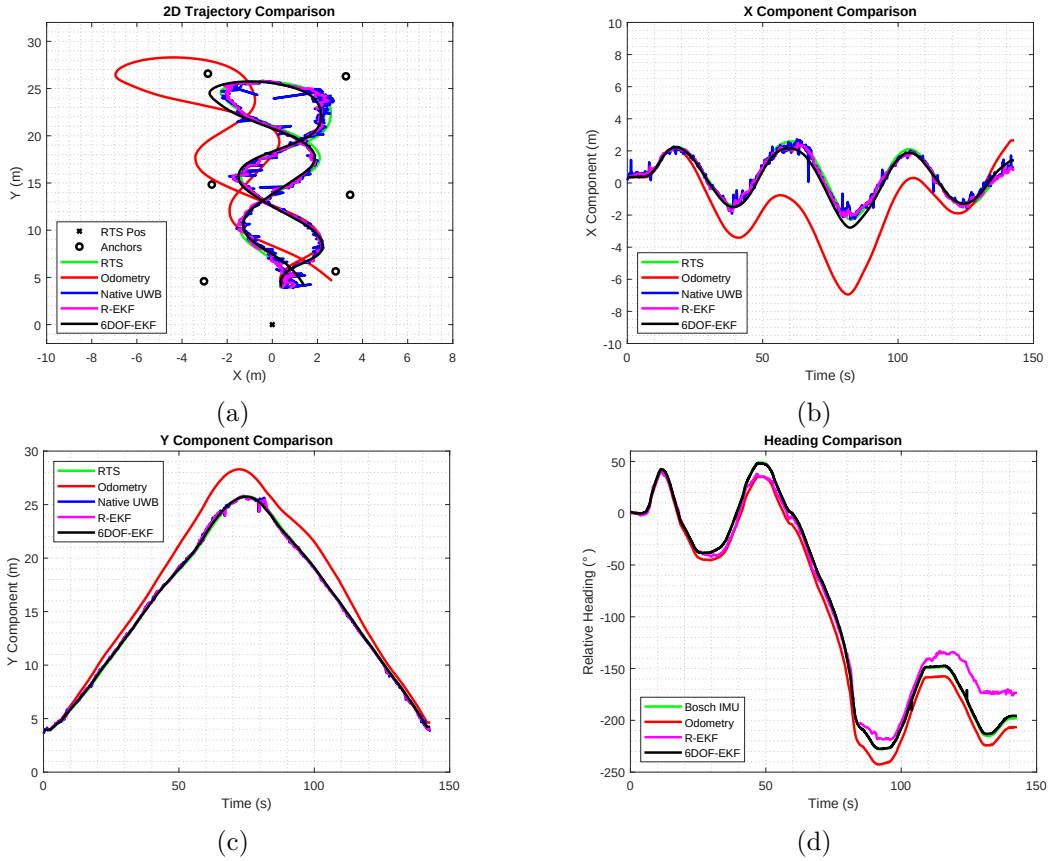


Figure 6.14: Components of motion comparison - Trajectory 4 (a) 2D Trajectory (b) Cartesian X (c) Cartesian Y (d) Heading

and accuracy due to the lack of external sensory information that can be deployed within a sensor fusion algorithm.

Table 6.7: Results Trajectory 4

Metric (m)	**Odometry**	**Native UWB**	**R-EKF**	**6DOF-EKF**
RMSE (x)	1.8031	0.2725	0.2096	0.2734
RMSE (y)	1.8994	0.2028	0.1943	0.0934
MBE (x)	1.8180	0.2009	0.1730	0.1535
MBE (y)	1.5443	0.1590	0.1539	0.0336

Table 6.8 presents the quantified error metrics concerning the heading estimations of each technique. It is a reminder here that the native UWB algorithm provides estimations concerning the Cartesian position only, therefore it is not included within the heading comparison results. The results presented in Table 6.8 show that out of the three techniques that were compared, the odometry estimations proved to yield the highest MBE, however,

it was the R-EKF which yielded the higher RMSE. This rise in RMSE for the R-EKF is judged to be a result of the final stages of the trajectory, where the R-EKF is seen not to converge in relation to the ground truth. This can be seen in the final 30 seconds within Figure 6.14d. Finally, Table 6.8 shows that the 6DOF-EKF heading estimations yielded the lowest overall RMSE and MBE, where both produced sub-degree error metrics.

Table 6.8: Heading Analysis Trajectory 4

Metric (∘)	**Odometry**	**R-EKF**	**6DOF-EKF**
RMSE	4.2334	14.28	0.8441
MBE	10.1048	5.3487	0.3661

6.6 Results Inclined Terrain

Section 6.6 presents the results acquired from the experimental procedure conducted in the environment possessing an inclined section of terrain. Section 6.6 presents the results acquired during the experimental testing of the Cartesian Z component and the rotation pitch angle of the 6DOF-EKF. The results for each component are provided in two separate sections below in Section 6.6.1 and Section 6.6.2.

6.6.1 Cartesian Z Component

Section 6.6.1 presents the results acquired from the analysis of the estimated Z component outputted from the 6DOF-EKF in comparison to the ground truth metric of the RTS. The RTS enables the ability to provide an accurate and noise free trajectory of the unmanned system concerning its change in the Z component in relation to its initial starting position. As presented in Section 6.4 concerning the methodological overview of the inclined terrain experimental procedure, four iterations of the experiment were undertaken to demonstrate repeatability. The results acquired from the inclined terrain experimental procedure are presented in Figure 6.15 for each iteration. It demonstrates that the inclination of the terrain that the unmanned system operates on, reaches a maximum relative altitude or maximum relative change in the Z component of motion by approximately 1m. This can be seen when observing the plot concerning the RTS. It is clear from observing Figures 6.15a to 6.15d that the estimations from the 6DOF-EKF are of a very similar nature to the

RTS. The 6DOF-EKF estimations for the Z component are comprised of a fusion between the unmanned system's estimated forward facing translational change within that certain time instance and the current estimation of the unmanned system's pitch angle. Therefore, the estimation outputs from the 6DOF-EKF for the Z component demonstrates low levels of noise or uncertainty within the estimation and when assessing the output graphically, maintains its accuracy in comparison to the RTS.

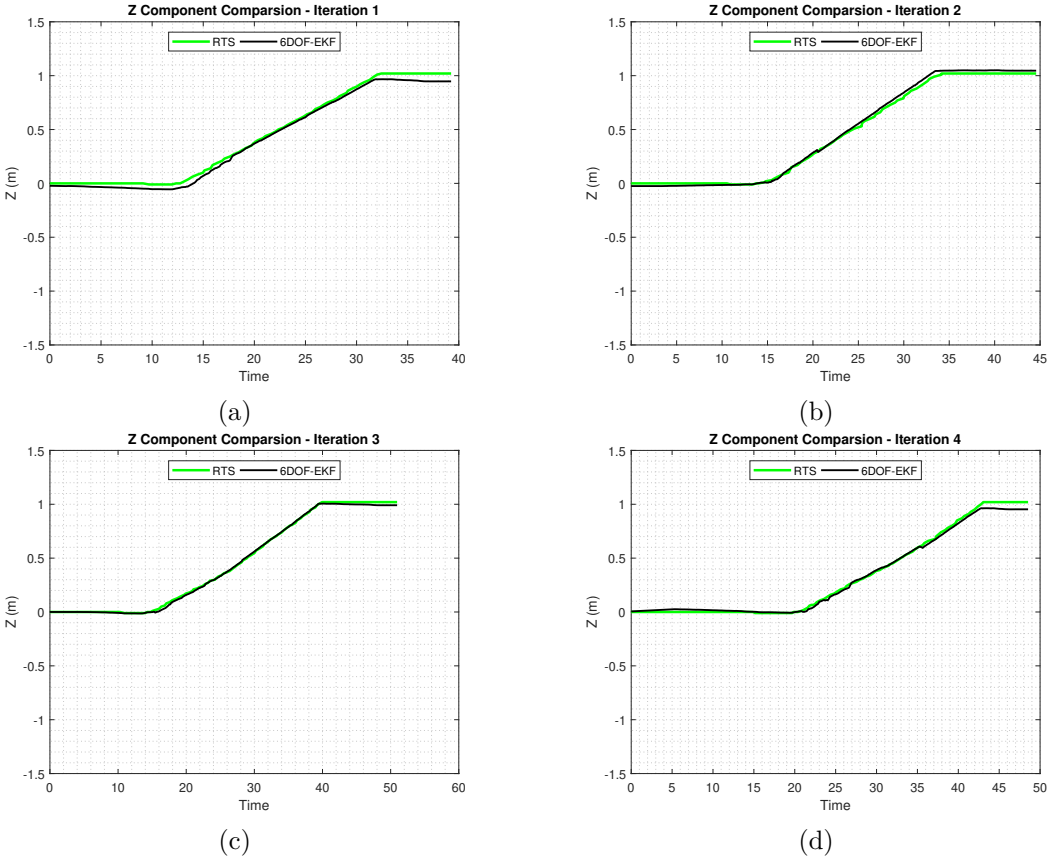


Figure 6.15: Z motion component comparison (a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4

Following on from the graphical representations of the results concerning the Z component of motion, Table 6.9 presents the resulting statistics in the form of the RMSE and MBE. For each iteration, the resulting RMSE was seen to be sub 5cm when using the RTS as the ground truth metric. These four iterations concluded with an average RMSE skill score of 0.0198m, which is approximately 1.98% of the total relative change of the Z component of motion. This process was also applied to the resulting MBE statistics. It can be seen from Table 6.9 that the resulting metrics for MBE for each iteration were also similar, however, iteration 2 produced a higher MBE in comparison to the other experimental iterations. The

overall mean MBE for all four iterations was calculated at 0.0214m, which is approximately 2.14% of the total relative change of the unmanned system's Cartesian Z component of motion during the inclined terrain experimental procedure.

Table 6.9: Cartesian Z Component Analysis

Metric (∘)	**Iteration No.**	**6DOF-EKF**
RMSE	1	0.0199
	2	0.0218
	3	0.0118
	4	0.0260
MBE	1	0.0184
	2	0.0363
	3	0.0103
	4	0.0205

Finally, Figure 6.16 shows the estimation error of the 6DOF-EKF in relation to time for each experimental iteration. This representation is included to demonstrate the temporal nature of the error and to identify areas where the error becomes more unstable in comparison to others. It can be seen when cross referencing each plot within Figure 6.16 with its corresponding plot in Figure 6.15 that when the unmanned system begins to proceed on the inclined terrain, the error within the estimation of Z becomes slightly more unstable. This can be seen when visually matching the time frames.

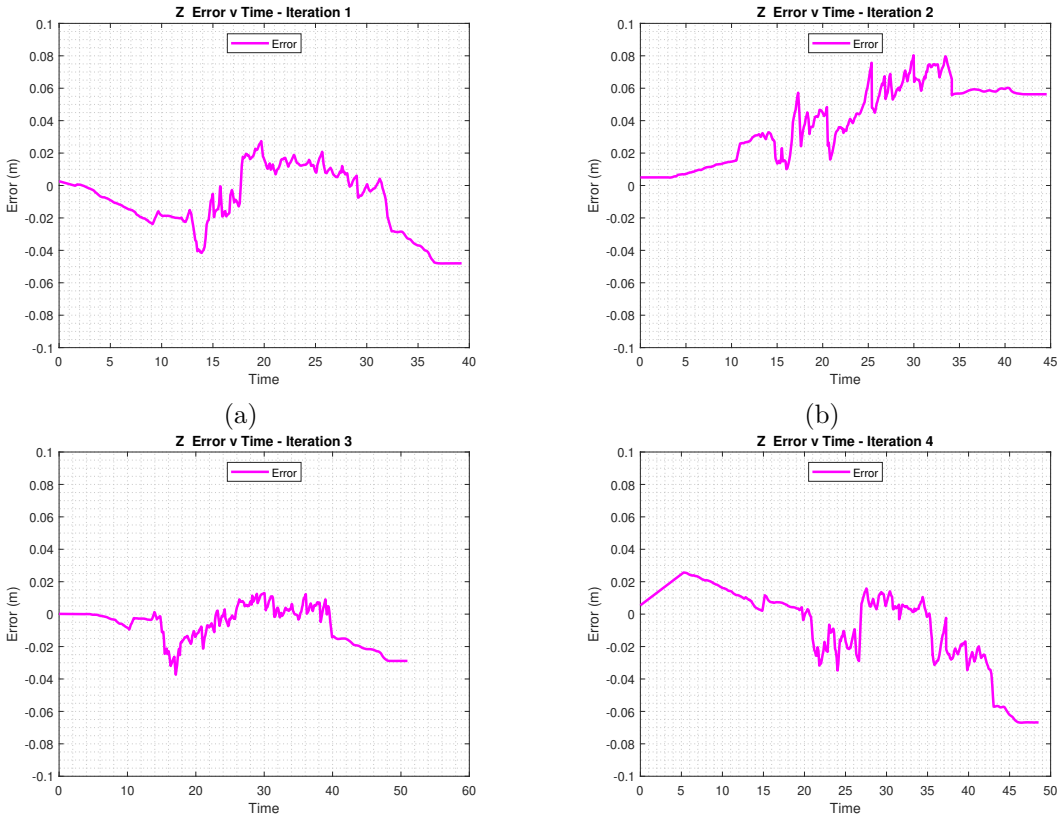


Figure 6.16: Z motion component estimation error (a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4

6.6.2 Rotational Pitch Component

Following on from Section 6.6 where the results concerning the Z component of motion were presented, Section 6.6.2 covers the results acquired for the rotational pitch angle change as the unmanned system embarked upon the inclined terrain. The unmanned system was equipped with the commercial grade Bosch IMU which provided a ground truth metric for quantitative comparison of the estimation algorithms.

The results obtained for pitch angle estimation in comparison to the Bosch IMU are demonstrated in Figure 6.17 for all four experimental iterations. Due to the inclined experimental procedure being conducted on the same terrain, it is evident from all four plots in Section 6.6.2 that the inclination of that certain section of terrain is approximately 9° as well as the areas where the terrain then becomes flat again, identified at the start and end of each experimental iteration. Figures 6.17a to 6.17d demonstrate visually the overall effectiveness of the pitch estimations produced by the 6DOF-EKF state estimation algorithm. It shows that the 6DOF-EKF estimations identify small and large changes in

angle over time when compared to that of the Bosch IMU, as well as generally following the plot produced by the ground truth metric. All four iterations are seen to behave in the same nature as the ground truth metric, however, it is identifiable that experimental iteration 2 visually produces the lowest accuracy in estimation, this is shown during the period of the iteration where the unmanned system is travelling up the inclined section of terrain where the 6DOF-EKF seems to overestimate the angle of inclination.

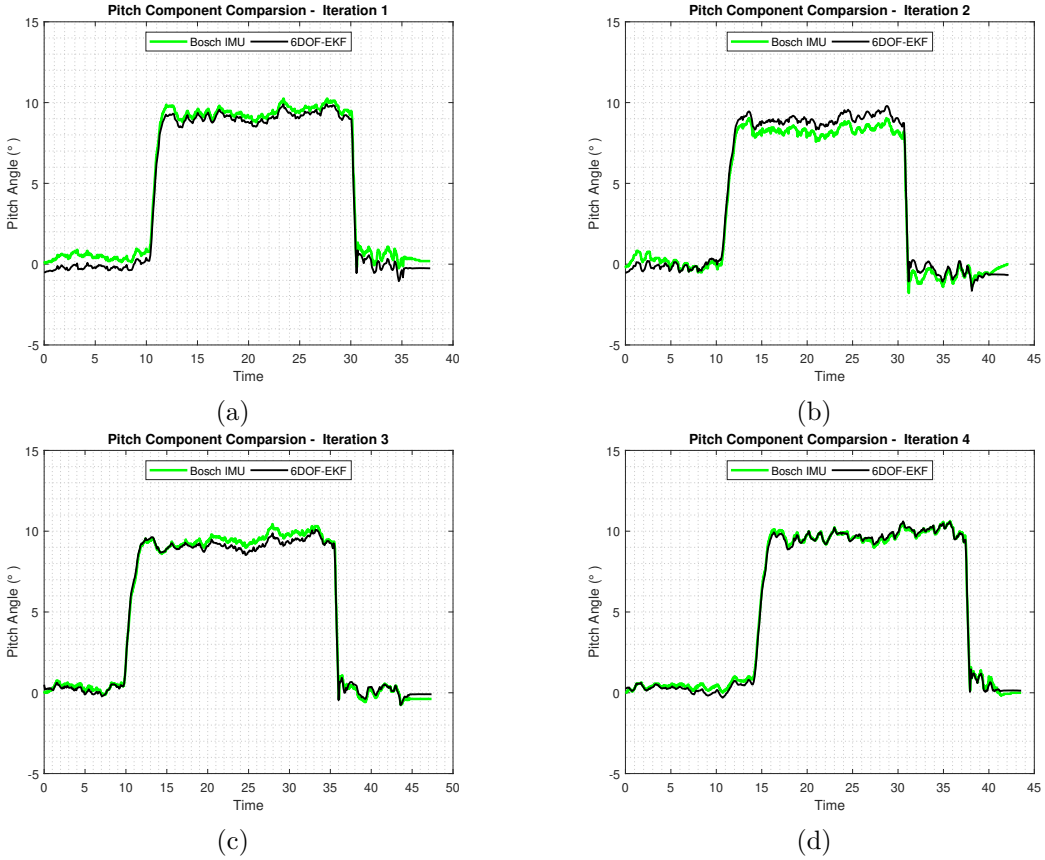


Figure 6.17: Pitch motion component comparison (a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4

In order to arrive at sound conclusions, error metrics were calculated to confirm what the graphical implementations in Figure 6.17 presented. Table 6.10 shows the numerical statistics in the form of the RMSE and MBE skill scores acquired from the inclined terrain experiment for the pitch rotational angle estimation. For experimental iterations 1, 3 and 4, the RMSE and MBE are seen to result below approximately 0.25°, which is approximately 2.6% of the maximum rotational change in pitch angle. However, iteration 2 produced slightly higher MBE and RMSE results, confirming what was demonstrated in Figure 6.17. The error statistics as a whole can be seen in Table 6.10.

Table 6.10: Pitch Component Analysis

Metric (∘)	**Iteration No.**	**6DOF-EKF**
RMSE	1	0.1854
	2	0.4500
	3	0.2532
	4	0.1671
MBE	1	0.1614
	2	0.8063
	3	0.2363
	4	0.1489

Finally, Figure 6.18 shows the temporal behaviour of the pitch error estimation for each experimental iteration. It shows that the error for iterations 1, 3 and 4 is fairly stable and low in magnitude. This however, is contrasting to that produced in iteration 2, where the error is seen to drift higher during the mid-parts of the procedure. This further confirms the claims produced in both Figure 6.17 and Table 6.10.

6.7 Scoring Table

In order to effectively summarise the results acquired in Chapter 6, a scoring table was designed to provide a finalised overview of the characteristics of each deployed technique. Characteristics such as RMSE and MBE are scored based on the presented results in Section 6.5 and Section 6.6. The ground truth metrics of the RTS and the Bosch IMU are included to show the characteristics of these techniques for features other than the components that they are providing a ground truth for. Additionally, 2D position and 2D pose are not equal. 2D position is defined as the 2D Cartesian co-ordinate in the form of (x,y) and 2D Pose includes the 2D position as well as the system's heading, which is in the form (x, y, ψ). The RTS functions very well providing 2D position, however, it lacks the ability to estimate the system's heading. This could possibly be achieved through the two RTS systems operating in conjunction. The Bosch IMU is scored the highest for its orientation estimation capabilities. However, it is scored at the lowest for its ability to provide Cartesian position estimations. This is due to recursive estimation

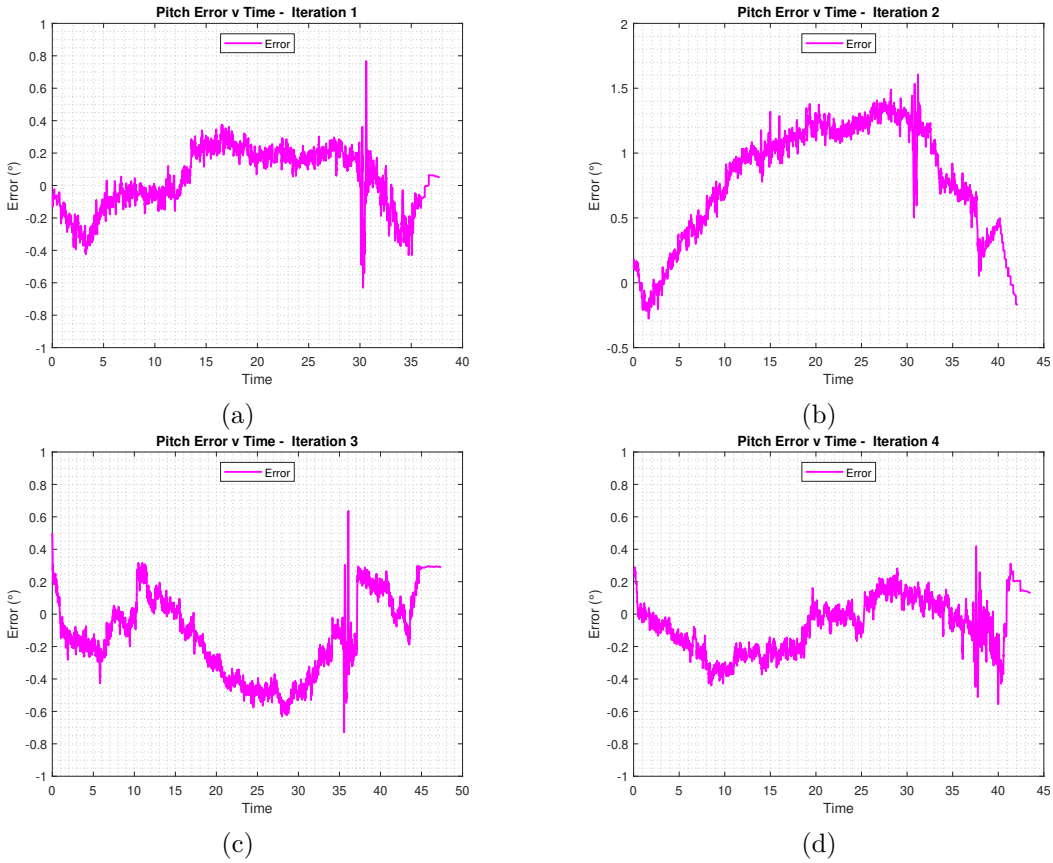


Figure 6.18: Pitch motion component estimation error (a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4

and removal of the forward translational acceleration that the filtering algorithm deploys in order to provide an optimal orientation estimation. Also, the temporal double integration of accelerometer readings that is necessary to achieve positional estimations experiences catastrophic error, this is due to the noisy nature of accelerometer outputs. As viewed from Table 6.11, the 6DOF-EKF was scored highest for all characteristics that were tested as well as its capabilities in comparison to the other techniques. However, one vital drawback of the 6DOF-EKF is that the LiDAR's performance as an observation source is heavily dependent upon the environment. This is the nature of the system and when situated within a feature rich environment, experiments have proven that the 6DOF-EKF is an effective solution for state estimation of an unmanned system. The solutions which include UWB measurements can operate in both feature rich and featureless environments as the UWB reference beacons can be placed within most operating environments. However, the performance of the solution operating using the Native UWB algorithm is dependent upon the accurate placement of the reference beacons within the environment. If the location of

each beacon in relation to the other beacons is incorrect, the final positioning estimation from the Native UWB algorithm is incorrect. This is due to the nature of the Linear Least Squares (LLS) technique that is deployed by the Native UWB algorithm. The R-EKF overcomes this by solely using raw range measurements to each beacon and dos not deploy an LLS multi-lateration technique. Finally, Table 6.11 also shows that the odometry scored the lowest out of the experimented solutions. Although the odometry trajectory estimations are visually more precise than the Native UWB and R-EKF, the effects due to wheel slippages and gyroscopic drift result in severe inaccuracies in estimations, deeming it a solution that is unsuitable to be deployed as a stand-alone.

Table 6.11: Scoring Table

Legend: V.Good - ***** Good - **** Standard - *** Poor - ** V.Poor - *

Characteristic	**Technique**					
	RTS	Bosch IMU	Odometry	Native UWB	R-EKF	6DOF-EKF
2D Position	*****	*	**	***	***	****
2D Pose	**	*	**	**	***	****
Full 3D Position	*****	*	*	**	*	***
Full 3D Orientation	*	*****	*	*	*	***
Full 3D Orientation Capability	*	*****	***	****	****	*****
MBE	*****	*****	*	**	***	****
RMSE	*****	*****	**	**	***	****
Heading Drift	*	*****	**	*	***	****
Environmentally Independent	*****	*****	*****	***	***	*
Line of Sight Independent	*	*****	*****	**	**	****
Total Score	**31**	**38**	**24**	**22**	**26**	**36**

6.8 Conclusions

In conclusion, Chapter 6 presented a series of experimental procedures that were undertaken to validate each technique viewed as a solution to indoor positioning. It presented a traditional odometry technique, a commercial off-the-shelf state of the art system and two custom probabilistic state estimation algorithms designed in Chapter 4 and Chapter 5. Each solution was compared to a ground truth metric in order to enable a quantitative analysis for error quantification. The conclusions made from the quantitative analysis and Chapter 6 as a whole are listed below;

- The use of an odometry system incorporating raw gyroscopic heading measurements and wheel encoders as a standalone system experiences catastrophic drift characteristics without the fusion of additional sensory information sources. Although this method severely lacks in accuracy, it produces a precise and noise-free trajectory estimation.

- The commercial UWB system known as Pozyx is an effective indoor localisation system. However, certain areas within the environment enclosed by the reference transceiver beacons hinder the precision of the provided position estimations.

- The implementation of a range-based EKF state estimation algorithm comprising of the odometry estimation as a control input, fused with raw UWB range measurements accompanied by a statistical outlier identification and mitigation procedure, demonstrates clear improvement in accuracy and precision in comparison to the commercial UWB technique.

- The integration of the odometry technique which is unsuitable to be deployed alone, can provide stabilisation effects to measurements acquired via UWB which are noisy by nature, to provide results which are more effective than a commercialised system. This demonstrates the effectiveness of probabilistic state estimations and sensor fusion implementations.

- The application of a 360° horizontal LiDAR scanning system as the heart of a state estimation process provides both accurate and precise estimations. The precision and control provided by a LiDAR proved to be more effective than the UWB solutions.

However, one major drawback concerning the LiDAR solution is that it needs a sufficient number of features within the environment in order to operate, therefore deeming the 6DOF-EKF severely environmentally dependent. This is a characteristic which does not apply to a UWB solution.

- The design of a 6DOF-EKF state estimation system, which comprises of two sub-systems can be an effective solution to estimating all 6 degrees of freedom of a rigid bodied system. The fusion of proprioceptive sensors such as accelerometers and gyroscopes with an indirect KF provides the ability to monitor aspects of rotational motion of the unmanned system, thus when in combination with the LiDAR, provides full 6DOF state estimation.

Part IV

Conclusions

Chapter 7

Conclusions & Future Work Recommendations

Chapter 7 is split into two final subsections which cover the conclusions drawn from this thesis and recommendations for future studies, assessment and experimentations to further enhance this research study.

7.1 Final Conclusions

A study which encompassed probabilistic and statistical theory, mathematical system modelling and unmanned system deployment was conducted with an aim to develop a mobile unmanned mapping system as a part of the EU Horizon 2020 research project known as DigiArt (Grant Agreement ID - 665066). DigiArt is comprised of 7 institutions spanning over 5 countries with an overall goal to virtualise active influential archaeological sites through remote sensing technology. A part of this project was to deploy unmanned systems to actively map the sites with a mobile LiDAR device, therefore the motion between each laser scan acquired from the unmanned system was required to be recursively tracked. To approach this goal, two customised algorithms known as the R-EKF and the 6DOF-EKF were designed, with both algorithms being a non-linear extended variant of the linear quadratic estimation algorithm known as the Kalman Filter. The general operational procedure of an EKF uses belief distributions to predict and update a priori and a posteriori estimations of the state of an unmanned system's components of motion. For the validation of both algorithms, an unmanned system is deployed whilst both the R-EKF and 6DOF-EKF operate on-board. Simultaneously, an optical measurement system in the form of a

Robotic Total Station (RTS) actively tracks the mobile system to provide a ground truth metric for quantitative analysis procedures.

The initial system assessed as a solution for position estimation of the unmanned system was a commercial ultra-wide band (UWB) network, where this UWB functioned through range estimations between a set of wireless beacons and a mobile tag on-board the unmanned system. Initial experiments into the deployment of the UWB network were conducted in, firstly, static conditions and secondly, in dynamic conditions. The static based experiments showed that the range estimation error between two static UWB nodes produced an average error of 10cm and 24cm in clear line of sight (CLOS) and non-clear line of sight (NCLOS) conditions respectively. However, the lowest average range error was acquired when the two UWB nodes' orientations were at a 90° angle, where the antennae on each device faced directly at each other. Such error metrics were calculated at 4.8cm and 11cm for CLOS and NCLOS conditions respectively. Testing the range estimation of the UWB system in dynamic conditions has shown that the system experienced a high number of range estimation outliers in comparison to that of the ground truth metric. This provided an initial insight into possible limitations of the commercial system and the effects of such outliers on the resulting position estimations, where outlier measurements were a result of the unmanned system operating in an area with less coverage from the static UWB nodes. Nonetheless, such outliers were identified and removed and the error residuals for all range UWB range measurements and results showed that an UWB system comprising of 7 nodes (6 static and 1 mobile) was found to have a generalised mean bias error (MBE) of 8.5cm, with a standard deviation of error at 19.72cm. It is to be noted, however, that the performance of the UWB system was affected when the unmanned system navigated to areas of the environment where the static UWB nodes did not provide enough coverage. Therefore, due to the nature of its operation, the state and condition of the environment is a significant factor to consider when deploying a wireless positioning system using UWB.

The observations acquired concerning the error in range estimations as well as occurrences of volatile measurements demonstrated that there were possible areas to improve the output of the UWB network. Therefore, an EKF was formulated and deployed using the raw range measurements and not the post calculated position estimation from the UWB system, this algorithm was denoted as the R-EKF. The R-EKF employed a Mahalanobis Distance (MD) metric to identify outlier measurements, and due to the

kinematic model of the system, it was able to estimate posterior states even in the absence of a sensor measurement update. Overall, integrating the acquired uncertainty within the range estimations from the UWB devices, live outlier detection and utilising a kinematic model of the unmanned system for prediction, the R-EKF outperformed the UWB system with an average error of 25cm compared to 40cm for X position and 13cm compared to 18cm for Y position. However, both the UWB and the R-EKF outperformed the traditional odometry approach. These results support the view that taking into consideration system kinematics, system inputs and sensor fusion implementations in the form of probabilistic estimators can improve accuracy and precision when relating to the topic of state estimation for mobile unmanned systems. However, it is the R-EKF and the UWB system only that provide position estimations in 2D.

This work was expanded to arrive at the design of the 6DOF-EKF, which provided estimations concerning both 3D position and orientation of the unmanned system. As opposed to the R-EKF, the 6DOF-EKF integrated a 360° LiDAR system as its primary measurement source, which provided motion estimations by assessing features in the environment. Firstly, the 2D pose was estimated using a normal distribution transformation (NDT) features matching process that operated on consecutive scans provided by the LiDAR and secondly, the final three components of motion to make full 3D pose estimation was initially achieved using ground plane recognition acquired from the LiDAR. However, experimentations proved that this approach failed when ground planes in two consecutive laser scans had an angle difference of 10.5°, as well as the strap-down method of mounting the laser scanning system to the unmanned platform proved a significant factor in the generation of accumulative estimation errors when in a static position.

Therefore, an approach was implemented to counter this limitation through the deployment of an Error-State Kalman Filter (ES-KF), which operated as a sub-system with the 6DOF-EKF. The ES-KF fused measurements from accelerometers and gyroscopes to estimate the roll and pitch angles by recursively estimating the location of the gravitational vector in relation to the unmanned system. Through this approach, the translational accelerations were able to be removed from accelerometer readings to enable roll and pitch angle calculation. The ES-KF was seen to compliment both accelerometer and gyroscopic readings to provide a smoother angle estimation as opposed to both sensors when operated as standalone.

Integrating the laser scanning system into the 6DOF-EKF proved that outlier measurements produced by the R-EKF as a result of the UWB were overcome. The laser scanning system as a measurement source proved to be a lot more stable as it does not rely on the CLOS to static beacons within the environment. However due to this lack of consistent and ever present reference points within the environment, the position estimations provided by the 6DOF-EKF were seen to slightly drift over time. Nonetheless, the experimental procedures conducted on the flat terrain showed the 6DOF-EKF outperformed other solutions including the R-EKF and the UWB for all 4 experimental trajectories, producing a less volatile trajectory estimation in comparison to the wireless radio based methods. The 6DOF-EKF outperformed the R-EKF for all results related to the Cartesian Y component with an average mean error of 12.13cm compared to 15.65cm of the R-EKF. This was also seen to outperform the native UWB system estimations which produced an error metric of 17.1cm. Concerning the Cartesian X component, the 6DOF-EKF outperformed the other solutions for 3 out of the 4 experimental trajectories, proving it is the ideal technique to deploy out of those explored.

Concerning the inclined terrain experiment, as a result of the 4 experimental iterations the 6DOF-EKF estimated the elevation of the unmanned system with an average error of 2.5cm when compared to the ground truth of the RTS. However, a significant factor in this estimation will be the strap down nature of the sensors on-board the unmanned system, as produced vibrations may effect estimations. Additionally, the 6DOF-EKF estimated the inclination angle as the unmanned system navigated up the experimental inclined terrain with an average error of 0.33° over the 4 experimental iterations.

Finally, all results were considered and a scoring table was generated considering the following factors:

- 2D Position
- 2D Pose
- Full 3D Position
- Full 3D Orientation
- Overall mean average error
- Standard deviation of error

- Heading drift
- Environmentally independent
- Line of sight independent

As a result of the aforementioned scoring mechanisms, the 6DOF-EKF had the highest score when compared to the other techniques explored, where such results were gained from the utilisation of an absolute ground truth metric.

7.2 Recommendations for Future Work

The presentation of novel literature, state of the art topics and the presentation of the findings acquired from experimental procedures conducted in this thesis focused on areas within robotics and general sensor implementations that few studies have been performed on. Therefore, Section 7.2 presents recommendations for future work to extend this topic of study.

In previous studies, the characterisation of uncertainty within range measurements between UWB nodes has only been considered through a generalisation approach, where the uncertainty metric is static. It is possible that the uncertainty metric may adapt based on the relationship of the unmanned systems position to the static UWB nodes. An example scenario of this is if the unmanned system navigated into an area within the environment that had less UWB coverage, therefore within this scenario an adaptive uncertainty metric may be an ideal solution.

If UWB is to be used as a reliable and effective positioning solution, considerations should be taken concerning the research into measurements acquired in NCLOS conditions. NCLOS between two communicating UWB devices can be detrimental to the final position estimation. It is recommended that future work focuses on identifying scenarios when there is an NCLOS between devices. Once an NCLOS condition is identified, the range measurement can be adjusted to produce overall position estimation error.

The 6DOF-EKF designed in Chapter 5 of this thesis should be developed into a full SLAM system to recursively recognise areas of the environment that it has observed before, through this approach loop closure processes can be implemented to optimise motion estimation and further reduce trajectory error.

The R-EKF designed in Chapter 4 and the 6DOF-EKF have been validated as individual working algorithms. For the purpose of further work, it is recommended that a blended system should be explored. Given the static behaviour of the UWB nodes within the environment, a fusion between LiDAR and UWB within a state estimation algorithm may reduce overall residual errors and optimise positioning for finite environments. However, this may only be suitable for explorations where the operating environment is constrained to the area covered by the static UWB nodes.

Other possible sensory systems that were not deployed in this study, such as vision-based devices are recommended to be explored and integrated within the algorithms designed in this thesis.

The implementation of novel and unique applications for autonomous and unmanned systems is growing and will continue to grow at a rapid rate. The ability to deploy intelligent systems to automate tasks and to provide in-depth information in environments in which humans cannot operate is becoming more increasingly required. This thesis presented the development and validation of control algorithms capable of being integrated with such systems to gather information on remote archaeological environments as a part of an EU research initiative. With the continuation of research and development projects, the industry for compact, robust and intelligent unmanned systems has the potential to grow and succeed.

Appendices

Appendix - ES-KF Process Noise Covariance Matrix & Correlation Terms

$$Q_k(\widehat{P}_{k-1}, \nu, \eta) = \begin{bmatrix} Q_{(1,1)} + \eta & 0 & 0 & -Q_{(1,4)} + \eta & 0 & 0 & 0 & 0 & 0 \\ 0 & Q_{(2,2)} + \eta & 0 & 0 & Q_{(2,5)} + \eta & 0 & 0 & 0 & 0 \\ 0 & 0 & Q_{(3,3)} + \eta & 0 & 0 & Q_{(3,6)} + \eta & 0 & 0 & 0 \\ -Q_{(4,1)} + \eta & 0 & 0 & Q_{(4,4)} + \eta & 0 & 0 & 0 & 0 & 0 \\ 0 & Q_{(5,2)} + \eta & 0 & 0 & Q_{(5,5)} + \eta & 0 & 0 & 0 & 0 \\ 0 & 0 & Q_{(6,3)} + \eta & 0 & 0 & Q_{(6,6)} + \eta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Q_{(7,7)} + \nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_{(8,8)} + \nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_{(9,9)} + \nu \end{bmatrix}$$

Autocorrelation Terms

- $Q_{(1,1)} = \widehat{P}_{k-1}(1,1) + \widehat{P}_{k-1}(4,4)$
- $Q_{(2,2)} = \widehat{P}_{k-1}(2,2) + \widehat{P}_{k-1}(5,5)$
- $Q_{(3,3)} = \widehat{P}_{k-1}(3,3) + \widehat{P}_{k-1}(6,6)$
- $Q_{(4,4)} = \widehat{P}_{k-1}(4,4)$
- $Q_{(5,5)} = \widehat{P}_{k-1}(5,5)$
- $Q_{(6,6)} = \widehat{P}_{k-1}(6,6)$
- $Q_{(7,7)} = \widehat{P}_{k-1}(7,7)$
- $Q_{(8,8)} = \widehat{P}_{k-1}(8,8)$
- $Q_{(9,9)} = \widehat{P}_{k-1}(9,9)$

Cross-correlation Terms

- $Q_{(1,4)} = \widehat{P}_{k-1}(4,4)$
- $Q_{(2,5)} = \widehat{P}_{k-1}(5,5)$
- $Q_{(3,6)} = \widehat{P}_{k-1}(6,6)$
- $Q_{(4,1)} = \widehat{P}_{k-1}(4,4)$
- $Q_{(5,2)} = \widehat{P}_{k-1}(5,5)$
- $Q_{(6,3)} = \widehat{P}_{k-1}(6,6)$

Appendix II - 6DOF-EKF Process Noise Covariance Matrix & Correlation Terms

$$Q_{6DOF-EKF} = \begin{bmatrix} Q_{(1,1)} & Q_{(1,2)} & Q_{(1,3)} & 0 & 0 & Q_{(1,6)} \\ Q_{(2,1)} & Q_{(2,2)} & Q_{(2,3)} & 0 & 0 & Q_{(2,6)} \\ Q_{(3,1)} & Q_{(3,2)} & Q_{(3,3)} & 0 & Q_{(3,5)} & 0 \\ 0 & 0 & 0 & Q_{(4,4)} & 0 & 0 \\ 0 & 0 & Q_{(5,3)} & 0 & Q_{(5,5)} & 0 \\ Q_{(6,1)} & Q_{(6,2)} & 0 & 0 & 0 & Q_{(6,6)} \end{bmatrix}$$

Autocorrelation Terms

- $Q_{(1,1)} = \frac{\sigma_\gamma^2 \cos\left(\psi + \frac{\dot{\psi}\, t}{2}\right)^2}{2} + \frac{{\sigma_{\dot{\psi}}}^2\, t^2 \sin\left(\psi + \frac{\dot{\psi}\, t}{2}\right)^2 \delta D^2}{4}$
- $Q_{(2,2)} = \frac{\sigma_\gamma^2 \sin\left(\psi + \frac{\dot{\psi}\, t}{2}\right)^2}{2} + \frac{{\sigma_{\dot{\psi}}}^2\, t^2 \cos\left(\psi + \frac{\dot{\psi}\, t}{2}\right)^2 \delta D^2}{4}$
- $Q_{(3,3)} = \frac{\sigma_\gamma^2 \sin\left(\varphi + \dot{\phi}\, t\right)^2}{2} + {\sigma_{\dot{\phi}}}^2\, t^2 \cos\left(\varphi + \dot{\phi}\, t\right)^2 \delta D^2$
- $Q_{(4,4)} = {\sigma_{\dot{\psi}}}^2\, t^2$
- $Q_{(5,5)} = {\sigma_{\dot{\phi}}}^2\, t^2$
- $Q_{(6,6)} = {\sigma_{\dot{\psi}}}^2\, t^2$

Cross-correlation Terms

- $Q_{(1,2)} = \frac{\sigma_\gamma^2 \cos\left(\psi + \frac{\dot{\psi}\, t}{2}\right) \sin\left(\psi + \frac{\dot{\psi}\, t}{2}\right)}{2} - \frac{{\sigma_{\dot{\psi}}}^2\, t^2 \cos\left(\psi + \frac{\dot{\psi}\, t}{2}\right) \sin\left(\psi + \frac{\dot{\psi}\, t}{2}\right) \delta D^2}{4}$
- $Q_{(1,3)} = \frac{\sigma_\gamma^2 \cos\left(\psi + \frac{\dot{\psi}\, t}{2}\right) \sin\left(\varphi + \dot{\phi}\, t\right)}{2}$
- $Q_{(1,6)} = -\frac{{\sigma_{\dot{\psi}}}^2\, t^2 \sin\left(\psi + \frac{\dot{\psi}\, t}{2}\right) \delta D}{2}$
- $Q_{(2,1)} = Q_{(1,2)}$
- $Q_{(2,3)} = \frac{\sigma_\gamma^2 \sin\left(\phi + \dot{\phi}\, t\right) \sin\left(\psi + \frac{\dot{\psi}\, t}{2}\right)}{2}$
- $Q_{(2,6)} = \frac{{\sigma_{\dot{\psi}}}^2\, t^2 \cos\left(\psi + \frac{\dot{\psi}\, t}{2}\right) \delta D}{2}$
- $Q_{(3,1)} = Q_{(1,3)}$
- $Q_{(3,2)} = Q_{(2,3)}$
- $Q_{(3,5)} = {\sigma_{\dot{\phi}}}^2\, t^2 \cos\left(\varphi + \dot{\phi}\, t\right) \delta D$
- $Q_{(5,3)} = Q_{(3,5)}$
- $Q_{(6,1)} = Q_{(1,6)}$
- $Q_{(6,2)} = Q_{(2,6)}$

Citations

[1] Brian Albright. *Essentials of mathematical statistics*. Jones & Bartlett Learning, 2014.

[2] Taro Yamane. "Statistics: An introductory analysis". In: (1973).

[3] Sheldon M Ross. *Introduction to probability and statistics for engineers and scientists*. Academic Press, 2014.

[4] Samarjit Das. *Time Series Analysis*. Princeton University Press, Princeton, NJ, 1994.

[5] Sheldon M Ross. *Introduction to probability models*. Academic press, 2014.

[6] Julius S Bendat and Allan G Piersol. *Random data: analysis and measurement procedures*. Vol. 729. John Wiley & Sons, 2011.

[7] Ronald E Walpole, Raymond H Myers, Sharon L Myers, and Keying Ye. *Probability and statistics for engineers and scientists*. Vol. 5. Macmillan New York, 1993.

[8] Timothy D Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[9] Peter Congdon. *Bayesian statistical modelling*. Vol. 704. John Wiley & Sons, 2007.

[10] Jose Bernado and Adrian Smith. *Bayesian Theory*. John Wiley & Sons, 2004.

[11] David Roxbee Cox and David Victor Hinkley. *Theoretical statistics*. Chapman and Hall/CRC, 1979.

[12] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[13] Moshe Kam, Xiaoxun Zhu, and Paul Kalata. "Sensor fusion for mobile robot navigation". In: *Proceedings of the IEEE* 85.1 (1997), pp. 108–119.

[14] Sebastian Thrun. "Probabilistic algorithms in robotics". In: *Ai Magazine* 21.4 (2000), pp. 93–93.

[15] Randall Smith, Matthew Self, and Peter Cheeseman. "Estimating uncertain spatial relationships in robotics". In: *Autonomous robot vehicles*. Springer, 1990, pp. 167–193.

[16] Lennart Ljung. "System identification". In: *Wiley Encyclopedia of Electrical and Electronics Engineering* (1999), pp. 1–19.

[17] Nicholas Rotella, Michael Bloesch, Ludovic Righetti, and Stefan Schaal. "State estimation for a humanoid robot". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2014, pp. 952–958.

[18] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. "A probabilistic approach to concurrent mapping and localization for mobile robots". In: *Autonomous Robots* 5.3-4 (1998), pp. 253–271.

[19] Roland Siegwart, Illah Reza Nourbakhsh, Davide Scaramuzza, and Ronald C Arkin. *Introduction to autonomous mobile robots*. MIT press, 2011.

[20] Dieter Fox, Jeffrey Hightower, Lin Liao, Dirk Schulz, and Gaetano Borriello. "Bayesian filtering for location estimation". In: *IEEE pervasive computing* 3 (2003), pp. 24–33.

[21] Julian Keilson. *Markov chain models—rarity and exponentiality*. Vol. 28. Springer Science & Business Media, 2012.

[22] Lawrence R Rabiner and Biing-Hwang Juang. "An introduction to hidden Markov models". In: *ieee assp magazine* 3.1 (1986), pp. 4–16.

[23] Kazufumi Ito. "Gaussian filter for nonlinear filtering problems". In: *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*. Vol. 2. IEEE. 2000, pp. 1218–1223.

[24] Fred Daum. "Nonlinear filters: beyond the Kalman filter". In: *IEEE Aerospace and Electronic Systems Magazine* 20.8 (2005), pp. 57–69.

[25] Mohinder S Grewal. *Kalman filtering*. Springer, 2011.

[26] Dan Simon. "Kalman filtering". In: *Embedded systems programming* 14.6 (2001), pp. 72–79.

[27] Michael Roth and Fredrik Gustafsson. "An efficient implementation of the second order extended Kalman filter". In: *14th International Conference on Information Fusion*. IEEE. 2011, pp. 1–6.

[28] Garry A Einicke and Langford B White. "Robust extended Kalman filtering". In: *IEEE Transactions on Signal Processing* 47.9 (1999), pp. 2596–2599.

[29] Mathieu St-Pierre and Denis Gingras. "Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system". In: *IEEE Intelligent Vehicles Symposium, 2004*. IEEE. 2004, pp. 831–835.

[30] Yao Wu, Tianmiao Wang, Jianhong Liang, Jiao Chen, Qiteng Zhao, Xingbang Yang, and Chenhao Han. "Experimental kinematics modeling estimation for wheeled skid-steering mobile robots". In: *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*. IEEE. 2013, pp. 268–273.

[31] Benjamin McLoughlin, Jeff Cullen, Andy Shaw, and Frederic Bezombes. "Towards an Unmanned 3D Mapping System Using UWB Positioning". In: *Annual Conference Towards Autonomous Robotic Systems*. Springer. 2018, pp. 416–422.

[32] Benjamin McLoughlin, Harry Pointon, John McLoughlin, Andy Shaw, and Frederic Bezombes. "Uncertainty Characterisation of Mobile Robot Localisation Techniques using Optical Surveying Grade Instruments". In: *Sensors* 18.7 (2018), p. 2274.

[33] Harry AG Pointon, Benjamin J McLoughlin, Christian Matthews, and Frederic A Bezombes. "Towards a Model Based Sensor Measurement Variance Input for Extended Kalman Filter State Estimation". In: *Drones* 3.1 (2019), p. 19.

[34] John J Leonard and Hugh F Durrant-Whyte. "Mobile robot localization by tracking geometric beacons". In: *IEEE Transactions on robotics and Automation* 7.3 (1991), pp. 376–382.

[35] Evgeni Kiriy and Martin Buehler. "Three-state extended kalman filter for mobile robot localization". In: *McGill University., Montreal, Canada, Tech. Rep. TR-CIM* 5 (2002), p. 23.

[36] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. "Robust Monte Carlo localization for mobile robots". In: *Artificial intelligence* 128.1-2 (2001), pp. 99–141.

[37] Teresa Conceição, Filipe Neves dos Santos, Paulo Costa, and António Paulo Moreira. "Robot Localization System in a Hard Outdoor Environment". In: *Iberian Robotics conference*. Springer. 2017, pp. 215–227.

[38] Chen Wang, Handuo Zhang, Thien-Minh Nguyen, and Lihua Xie. "Ultra-wideband aided fast localization and mapping system". In: *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE. 2017, pp. 1602–1609.

[39] Mark W Mueller, Michael Hamer, and Raffaello D'Andrea. "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation". In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 1730–1736.

[40] Thien Minh Nguyen, Abdul Hanif Zaini, Kexin Guo, and Lihua Xie. "An ultra-wideband-based multi-UAV localization system in GPS-denied environments". In: *Proc. Int. Micro Air Vehicle Conf. Competition*. 2016, pp. 1–6.

[41] Yang Song, Mingyang Guan, Wee Peng Tay, Choi Look Law, and Changyun Wen. "UWB/LiDAR Fusion For Cooperative Range-Only SLAM". In: *arXiv preprint arXiv:1811.02854* (2018).

[42] Joseph Djugash, Sanjiv Singh, George Kantor, and Wei Zhang. "Range-only slam for robots operating cooperatively with sensor networks". In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE. 2006, pp. 2078–2084.

[43] Jose-Luis Blanco, Javier González, and Juan-Antonio Fernández-Madrigal. "A pure probabilistic approach to range-only SLAM." In: *ICRA*. Citeseer. 2008, pp. 1436–1441.

[44] Geovanni Martinez. "Field tests on flat ground of an intensity-difference based monocular visual odometry algorithm for planetary rovers". In: *Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on*. IEEE. 2017, pp. 161–164.

[45] Craig Roberts and Peter Boorer. "Kinematic positioning using a robotic total station as applied to small-scale UAVs". In: *Journal of Spatial Science* 61.1 (2016), pp. 29–45.

[46] Artyom Maxim, Otto Lerke, Marshall Prado, Moritz Dörstelmann, Achim Menges, and Volker Schwieger. "UAV Guidance with Robotic Total Station for Architectural Fabrication Processes". In: *Unmanned Aerial Vehicles, Wißner-Verlag, Augsburg* (2017), pp. 145–161.

[47] G Roberto Aiello and Gerald D Rogerson. "Ultra-wideband wireless systems". In: *IEEE microwave magazine* 4.2 (2003), pp. 36–47.

[48] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. "Survey of wireless indoor positioning techniques and systems". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.6 (2007), pp. 1067–1080.

[49] Sinan Gezici, Zhi Tian, Georgios B Giannakis, Hisashi Kobayashi, Andreas F Molisch, H Vincent Poor, and Zafer Sahinoglu. "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks". In: *IEEE signal processing magazine* 22.4 (2005), pp. 70–84.

[50] Robert J Fontana. "Recent system applications of short-pulse ultra-wideband (UWB) technology". In: *IEEE Transactions on microwave theory and techniques* 52.9 (2004), pp. 2087–2104.

[51] *Pozyx - Accurate Positioning*. `https://www.pozyx.io/`. Accessed: 2018-10-25.

[52] Peng Rong and Mihail L Sichitiu. "Angle of arrival localization for wireless sensor networks". In: *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on*. Vol. 1. IEEE. 2006, pp. 374–382.

[53] Dragos Niculescu and Badri Nath. "Ad hoc positioning system (APS) using AOA". In: *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. Vol. 3. IEEE. 2003, pp. 1734–1743.

[54] Fazeelat Mazhar, Muhammad Gufran Khan, and Benny Sällberg. “Precise Indoor Positioning Using UWB: A Review of Methods, Algorithms and Implementations”. In: *Wireless Personal Communications* 97.3 (2017), pp. 4467–4491.

[55] Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Al-Hadhrami, Mai A. Al-Ammar, and Hend S. Al-Khalifa. “Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances”. In: *Sensors* 16.5 (2016). ISSN: 1424-8220. DOI: 10.3390/s16050707. URL: http://www.mdpi.com/1424-8220/16/5/707.

[56] Ruei-Shiue Shiu, Shih-Chung Kang, Jen-Yu Han, and Shang-Hsien Hsieh. “Modeling systematic errors for the angle measurement in a virtual surveying instrument”. In: *Journal of Surveying Engineering* 137.3 (2011), pp. 81–90.

[57] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. “ROS: an open-source Robot Operating System”. In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe, Japan. 2009, p. 5.

[58] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. “The mahalanobis distance”. In: *Chemometrics and intelligent laboratory systems* 50.1 (2000), pp. 1–18.

[59] Xun S Zhou and Stergios I Roumeliotis. “Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case”. In: *2006 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2006, pp. 1785–1792.

[60] *Boston Dynamics - Changing your idea of what robots can do.* https://www.bostondynamics.com/. Accessed: 2019-01-07.

[61] *Boston Dynamics Atlas - The World’s Most Dynamic Humanoid.* https://www.bostondynamics.com/atlas. Accessed: 2019-01-07.

[62] Jeremy Ma, Sara Susca, Max Bajracharya, Larry Matthies, Matt Malchano, and Dave Wooden. “Robust multi-sensor, day/night 6-DOF pose estimation for a dynamic legged vehicle in GPS-denied environments”. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, pp. 619–626.

[63] *Boston Dynamics - SpotMini - Good Things Come in Small Packages.* https://www.bostondynamics.com/spot-mini/. Accessed: 2019-01-07.

[64] A. Z. Zhu, N. Atanasov, and K. Daniilidis. “Event-Based Visual Inertial Odometry”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5816–5824. DOI: 10.1109/CVPR.2017.616.

[65] Mary B Alatise and Gerhard P Hancke. “Pose estimation of a mobile robot based on fusion of IMU data and vision data using an extended kalman filter”. In: *Sensors* 17.10 (2017), p. 2164.

[66] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. “Speeded-up robust features (SURF)”. In: *Computer vision and image understanding* 110.3 (2008), pp. 346–359.

[67] Jeroen D Hol, Fred Dijkstra, Henk Luinge, and Thomas B Schon. “Tightly coupled UWB/IMU pose estimation”. In: *Ultra-Wideband, 2009. ICUWB 2009. IEEE International Conference on*. IEEE. 2009, pp. 688–692.

[68] Jakub Simanek, Michal Reinstein, and Vladimir Kubelka. “Evaluation of the EKF-based estimation architectures for data fusion in mobile robots”. In: *IEEE/ASME Transactions on Mechatronics* 20.2 (2015), pp. 985–990.

[69] Ji Zhang and Sanjiv Singh. “LOAM: Lidar Odometry and Mapping in Real-time.” In: *Robotics: Science and Systems*. Vol. 2. 2014, p. 9.

[70] Ji Zhang and Sanjiv Singh. “Visual-lidar odometry and mapping: Low-drift, robust, and fast”. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 2174–2181.

[71] *KAARTA - Rapid and Accurate Mobile 3D Scan Technology*. http://www.kaarta.com/. Accessed: 2018-10-29.

[72] Gaurav Pandey, Shashank Giri, and Jame R Mcbride. "Alignment of 3D point clouds with a dominant ground plane". In: *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE. 2017, pp. 2143–2150.

[73] Hugh Durrant-Whyte and Tim Bailey. "Simultaneous localization and mapping: part I". In: *IEEE robotics & automation magazine* 13.2 (2006), pp. 99–110.

[74] David M Cole and Paul M Newman. "Using laser range data for 3D SLAM in outdoor environments". In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE. 2006, pp. 1556–1563.

[75] Jan Weingarten and Roland Siegwart. "EKF-based 3D SLAM for structured environment reconstruction". In: *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE. 2005, pp. 3834–3839.

[76] Michael Bosse, Robert Zlot, and Paul Flick. "Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping". In: *IEEE Transactions on Robotics* 28.5 (2012), pp. 1104–1119.

[77] *GEOSLAM*. https://geoslam.com/. Accessed: 2018-10-29.

[78] *Vicon Motion Tracking Systems*. https://www.vicon.com/. Accessed: 2018-11-05.

[79] Pierre Merriaux, Yohan Dupuis, Rémi Boutteau, Pascal Vasseur, and Xavier Savatier. "A Study of Vicon System Positioning Performance". In: *Sensors* 17.7 (2017). ISSN: 1424-8220. DOI: 10.3390/s17071591. URL: http://www.mdpi.com/1424-8220/17/7/1591.

[80] Robert J Aughey. "Applications of GPS technologies to field sports". In: *International journal of sports physiology and performance* 6.3 (2011), pp. 295–310.

[81] Rob Duffield, Machar Reid, John Baker, and Wayne Spratford. "Accuracy and reliability of GPS devices for measurement of movement patterns in confined spaces for court-based sports". In: *Journal of Science and Medicine in Sport* 13.5 (2010), pp. 523–525.

[82] Angelo M Sabatini. "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing". In: *IEEE Transactions on Biomedical Engineering* 53.7 (2006), pp. 1346–1356.

[83] Ling Chen, Huosheng Hu, and Klaus McDonald-Maier. "Ekf based mobile robot localization". In: *Emerging Security Technologies (EST), 2012 Third International Conference on*. IEEE. 2012, pp. 149–154.

[84] Shaojie Shen, Nathan Michael, and Vijay Kumar. "Autonomous multi-floor indoor navigation with a computationally constrained MAV". In: *Robotics and automation (ICRA), 2011 IEEE international conference on*. IEEE. 2011, pp. 20–25.

[85] Friedrich Keller and Harald Sternberg. "Multi-sensor platform for indoor mobile mapping: System calibration and using a total station for indoor applications". In: *Remote sensing* 5.11 (2013), pp. 5805–5824.

[86] Ablie Jarvis. *Advancing Trucking Safety with LiDAR*. 2018. URL: https://velodynelidar.com/newsroom/advancing-trucking-safety-with-lidar/ (visited on 10/08/2018).

[87] ND Quadros, PA Collier, and CS Fraser. "Integration of bathymetric and topographic LiDAR: a preliminary investigation". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 36 (2008), pp. 1299–1304.

[88] Ievgeniia Maksymova, Christian Steger, and Norbert Druml. "Review of LiDAR Sensor Data Acquisition and Compression for Automotive Applications". In: *Multidisciplinary Digital Publishing Institute Proceedings*. Vol. 2. 13. 2018, p. 852.

[89] Thomas PF Dowling, Helena Alexanderson, and Per Möller. "The new high-resolution LiDAR digital height model ('Ny Nationell Höjdmodell') and its application to Swedish Quaternary geomorphology". In: *Gff* 135.2 (2013), pp. 145–151.

[90] Gérard Lachapelle. "High sensitivity GNSS limitations in RF perturbed environments". In: *NATO STO lecture series SET-197, navigation sensors and systems in GNSS degraded and denied environments* (2013).

[91] Pavel Puricer and Pavel Kovar. "Technical limitations of GNSS receivers in indoor positioning". In: *2007 17th International Conference Radioelektronika*. IEEE. 2007, pp. 1–5.

[92] Velodyne. *VLP-16 User Manual*. English. Version 63-9243 Revision D. Velodyne LiDAR. 138 pp.

[93] Direct Dimensions. *FARO Laser Scanner Photon 120/20 Data Sheet*. English. Version Revision Code: 04REF201-061. FARO. 138 pp.

[94] Mathieu Dassot, Thiéry Constant, and Meriem Fournier. "The use of terrestrial LiDAR technology in forest science: application fields, benefits and challenges". In: *Annals of forest science* 68.5 (2011), pp. 959–974.

[95] Nicolas Brodu and Dimitri Lague. "3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 68 (2012), pp. 121–134.

[96] Joon Heo, Seongsu Jeong, Hyo-Keun Park, Jaehoon Jung, Soohee Han, Sungchul Hong, and Hong-Gyoo Sohn. "Productive high-complexity 3D city modeling with point clouds collected from terrestrial LiDAR". In: *Computers, Environment and Urban Systems* 41 (2013), pp. 26–38.

[97] Yongtao Yu, Jonathan Li, Haiyan Guan, Cheng Wang, and Jun Yu. "Semiautomated extraction of street light poles from mobile LiDAR point-clouds". In: *IEEE Transactions on Geoscience and Remote Sensing* 53.3 (2015), pp. 1374–1386.

[98] Bhavesh Kumar, Gaurav Pandey, Bharat Lohani, and Subhas C Misra. "A multi-faceted CNN architecture for automatic classification of mobile LiDAR data and an algorithm to reproduce point cloud samples for enhanced training". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 147 (2019), pp. 80–89.

[99] Pablo Rodríguez-Gonzálvez, Belén Jiménez Fernández-Palacios, Ángel Luis Muñoz-Nieto, Pedro Arias-Sanchez, and Diego Gonzalez-Aguilera. "Mobile LiDAR system: new possibilities for the documentation and dissemination of large cultural heritage sites". In: *Remote Sensing* 9.3 (2017), p. 189.

[100] S Khan, L Aragão, and J Iriarte. "A UAV–lidar system to map Amazonian rainforest and its ancient landscape transformations". In: *International journal of remote sensing* 38.8-10 (2017), pp. 2313–2330.

[101] Ji Zhang and Sanjiv Singh. "Aerial and Ground-Based Collaborative Mapping: An Experimental Study". In: *Field and Service Robotics*. Springer. 2018, pp. 397–412.

[102] Wai Yeung Yan, Ahmed Shaker, and Nagwa El-Ashmawy. "Urban land cover classification using airborne LiDAR data: A review". In: *Remote Sensing of Environment* 158 (2015), pp. 295–310.

[103] Jian S Dai. "Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections". In: *Mechanism and Machine Theory* 92 (2015), pp. 144–152.

[104] Peter Biber and Wolfgang Straßer. "The normal distributions transform: A new approach to laser scan matching". In: *Intelligent Robotics (IROS), 2003 IEEE International conference on, volume=3, pages=2743–2748, year=2003 organization=IEEE/RSJ*.

[105] Mohammad Shabani Sheijani, Asghar Gholami, Narjes Davari, and Mehdi Emami. “Implementation and performance comparison of indirect Kalman filtering approaches for AUV integrated navigation system using low cost IMU”. In: *2013 21st Iranian Conference on Electrical Engineering (ICEE)*. IEEE. 2013, pp. 1–6.

[106] Joan Sola. “Quaternion kinematics for the error-state Kalman filter”. In: *arXiv preprint arXiv:1711.02508* (2017).

[107] Nikolas Trawny and Stergios I Roumeliotis. “Indirect Kalman filter for 3D attitude estimation”. In: *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep* 2 (2005), p. 2005.